VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SMĚROVÁNÍ V DATOVÝCH SÍTÍCH
ROUTING PRINCIPLES IN DATA NETWORKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Tomáš Stodůlka**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Vít Novotný, Ph.D.**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Tomáš Stodůlka ***ID:*** 174399
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Směrování v datových sítích

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku směrování v datových sítích - statické a dynamické způsoby. Na základě nabytých znalostí a dostupného vybavení navrhněte laboratorní úlohu pro předmět Architektura sítí, sestavte a zprovozněte pracoviště a k úloze vypracujte návod.

DOPORUČENÁ LITERATURA:

[1] BLACK, Uyless D. IP routing protocols: RIP, OSPF, BGP, PNNI, and Cisco routing protocols. Upper Saddle River, NJ: Prentice Hall, 2000. ISBN 0130142484.

[2] DISCHER,S.R.W. RouterOS by Example. Stephen R.W. Discher, ISBN 978-0-615-54704-6, USA, 2011

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 8.6.2017

Vedoucí práce: doc. Ing. Vít Novotný, Ph.D.
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato bakalářská práce se zabývá problémem směrování v datových sítích. Teoretická část popisuje obecné možnosti spojení mezi dvěma uzly. Dále jsou popsány principy přepínání v rámci jedné sítě a směrování mezi sítěmi. Další část práce se zaměřuje na metody směrování z hlediska statického či dynamického zápisu do směrovací tabulky. V praktické části byla navržena a realizována struktura a topologie laboratorní úlohy pro předmět Architektura sítí.

KLÍČOVÁ SLOVA

Směrovač, výběr cesty, statické směrování, dynamické směrování, RIP, OSPF.

ABSTRACT

This bachelor thesis deals with the problem of routing in data networks. The theoretical part defines general options of connections between two nodes. Furthermore, the principles of switching within a single network and routing between the networks are described. Another part of thesis focuses on the routing methods in terms of static or dynamic enrolment into the routing table. In the practical part have been designed and realized a structure and topology of laboratory exercise for subject Architecture of networks.

KEYWORDS

Router, route selection, static routing, dynamic routing, RIP, OSPF.

STODŮLKA, Tomáš. *Směrování v datových sítích*. Brno, 2017, 76 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Vít Novotný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Směrování v datových sítích“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Vítu Novotnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM VÝPISŮ

ÚVOD

Globální pokrytí síťovou infrastrukturou je v dnešní době nezbytným základem pro většinu zařízení, ať už se jedná o stolní počítače, chytré mobilní telefony, hodinky, GPS navigace či televizory. Postupem času se systém propojení těchto zařízení stále vyvíjel a pro jeho správné fungování byly kladeny čím dál vyšší nároky, tedy z malých počítačových sítí se stávaly mnohem větší a bylo zapotřebí nějakým řešením určit způsob doručení dat od zdroje k cíli, ke kterému vede mnoho různých cest napříč síťovými uzly. K určení té nejlepší cesty slouží směrování.

Cílem této práce je popsat přenos informací od zdroje k cíli a dále se soustředit na principy směrování informací pomocí různých směrových protokolů. Na základě těchto poznatků bude navrhnuta a realizována laboratorní úloha pro předmět Architektura sítí, která studentům pomůže pochopit tento složitý proces směrování.

První kapitola popisuje teoretickou část bakalářské práce. Ze začátku se věnuje propojením uzlů v síti a možnosti určení cesty od zdroje k cíli v souvislosti s vrstvami modelu OSI. Dále se v kapitole rozebírá určování cesty k cíli v rámci jedné sítě (přepínání) a mezi sítěmi (směrování). Do větší míry tato kapitola přibližuje statické a dynamické metody směrování. Důraz je kladen hlavně na algoritmy dynamického směrování a směrové protokoly RIP a OSPF. **Druhá kapitola** rozebírá návrh laboratorní úlohy. Jsou zde vymezeny požadavky a cíle, kterými se řídit v nadcházející realizaci úlohy a dále stručný popis prvků, které budou v úloze využívány. **Třetí kapitola** se věnuje přípravě pracoviště ze strany jak počítačů, tak i směrovačů. Rovněž je zde pojednáváno o problematice, se kterou se autor této práce setkal v průběhu řešení. Na závěr je vypracován návod pro laboratorní úlohu předmětu Architektura sítí.

1 TEORETICKÁ ČÁST STUDENTSKÉ PRÁCE

1.1 Přepojování datových toků

Každá telekomunikační síť sestává z koncových uzlů, kde se realizují aplikace, jež jsou zdrojem, spotřebičem či úložištěm dat. Aby libovolné koncové uzly mohly spolu komunikovat za účelem výměny informací, musí být vzájemně propojeny. To může být realizováno systémem spojení „každý s každým“, avšak toto je uskutečnitelné pouze pro malý počet koncových uzlů z důvodu požadavku na velký počet spojů či z důvodu malého dosahu, nízké výsledné propustnosti a komplikovanosti řízení u sběrnicového řešení sítě. Typickým řešením ve velkém měřítku je realizace komplexní sítě obsahující struktury různě propojených přepojovacích uzlů, které přepojují data od zdroje směrem k cíli. Cesta mezi dvěma uzly může být v telekomunikační síti určena:

- **před vlastním přenosem dat:**
  - na fyzické vrstvě – spojování fyzických okruhů, vyhrazení fyzických komunikačních kanálů napříč celou sítí:
    - trvale – permanentní fyzické okruhy,
    - na dobu realizace služby – komutace fyzických okruhů,
  - virtuálně – vyhledání vhodné cesty a vytvoření záznamů v přepojovacích tabulkách vytyčujících trasu,
- **v průběhu přepojování:**
  - na linkové vrstvě – na základě přepínací tabulky přepínače,
  - na síťové vrstvě – na základě směrovací tabulky směrovačů,
  - na MPLS vrstvě – na bázi tabulek návěští,
  - na transportní vrstvě – na bázi přepojovací tabulky dle portů,
  - na aplikační vrstvě – na bázi přepojovací tabulky požadavků.

1.1.1 Referenční model ISO/OSI

Dříve než bude vysvětlen princip přepínání a směrování, jsou zde připomenuty základy referenčního modelu OSI (Open Systems Interconnection) vyvinutého Mezinárodní organizací pro normalizaci ISO. Model tvoří 7 vrstev (zkráceně se označují jako L1 – L7), kde každá plní v síťové komunikaci určitou úlohu [14], [16], [17].

- (L1) **Fyzická vrstva** – Přenáší tok bitů na tok symbolů a zabývá se, jakým způsobem jsou tyto symboly modulovány přes fyzické médium.
- (L2) **Linková (spojová) vrstva** – Vytváří rámce, které adresuje v dané síti

na základě MAC adres[1]. Řídí tok dat (čímž se předchází zahlcení) a chybové stavy.

- (L3) **Síťová vrstva** – Zajišťuje hlavně směrování a přenos datových jednotek (paketů) od vysílače k přijímači na základě cílové IP adresy[2].
- (L4) **Transportní vrstva** – Zajišťuje komunikaci mezi dvěma koncovými uzly pomocí spojově či nespojově orientovaného protokolu. Rozpoznává názvy koncových zařízení a přiřazuje je k logickým adresám. Segmentuje a znovu skládá data.
- (L5) **Relační vrstva** – Organizuje a synchronizuje dialog (navázání, udržení a ukončení) mezi aplikačními protokoly.
- (L6) **Prezentační vrstva** – Stará se o zdrojové kódování, případně o šifrování, či kompresi přenášených dat. Obecně řeší problém, že stejný řetězec bitů může mít pro příjemce jiný význam než pro odesílatele.
- (L7) **Aplikační vrstva** – Poslední vrstva OSI, která zajišťuje vzájemnou spolupráci mezi aplikačním programem a sítí. Neobsahuje aplikace, ale komunikuje s nimi.

1.2 Přepínání

Každá vrstva modelu ISO/OSI rozumí pouze informacím na své vrstvě, proto fyzická vrstva nerozumí datům obsaženým v rámci a tudíž nezajišťuje přepínání dat dle cílové adresy. Nejčastější zařízení fyzické vrstvy je rozbočovač, nebo opakovač, jejich zastoupení je dnes však minimální. Při obdržení zprávy je bez jakéhokoli zdržení přepojena na všechny ostatní výstupní porty.

Přepínání (Switching) je adresace datových jednotek již na vyšší vrstvě (L2), která rozumí obsahu rámce na linkové vrstvě a tudíž dokáže přepínat rámce určenému adresátovi. O přepínání se starají aktivní prvky druhé vrstvy ISO/OSI, nejčastěji se jedná o zařízení zvaná přepínače a mosty.

1.2.1 Princip přepínání

Přepínače rozesílají provoz v lokálních sítích (LAN) za pomoci přepínací tabulky (velmi rychlá vyrovnávací pamět typu CAM nebo TCAM). Příklad takové tabulky zobrazuje obrázek 1.1, primárně obsahuje **MAC adresy** různých zařízení, **typ** zápisu a výstupní **port**. Typ může být dynamický (údaj byl získán na základě příchozího rámce) nebo statický (zapsaný administrátorem). Dynamický údaj má navíc

[1]Fyzická adresa zařízení tvořena 48 bitovým unikátním číslem. Slouží k identifikaci zařízení v síti na linkové vrstvě [17].

[2]Je 32 nebo 128 bitová adresa identifikující síť a adresu hostitele, kterou používá IP [17].

dobu, po jejímž uplynutí údaj zaniká (tato doba se liší v závislosti na používaném operačním systému, například u přepínačů Cisco je tomu standardně 240 minut, u OS Windows zaniká za 10 minut). Pokud zařízení podporuje správu virtuálních lokálních sítí, tabulka bude obsahovat i **VLAN ID** (identifikační číslo virtuálních LAN) [8]. Dynamický zápis začíná například příchodem rámce na přepínač, ten si

Vlan ID	MAC adresa	Typ	Port
1	00.02.16.c0.41.01	dynamic	Fa0/3
1	00.02.17.98.d4.34	dynamic	Fa0/1
1	00.30.a3.dc.e0.d4	dynamic	Fa0/2
1	00.d0.58.c6.4a.a7	dynamic	Fa0/2
1	00.d0.97.73.1e.01	dynamic	Fa0/2

Obr. 1.1: Příklad tabulky přepínače

zapíše zdrojovou MAC adresu a přiřadí ji k portu, odkud rámec přišel. Tyto údaje si uloží do přepínací tabulky. Po uložení záznamu se podívá na cílovou MAC adresu a začne ji porovnávat s adresami v přepínací tabulce. Pokud ji najde, odešle rámec příslušným portem, pokud záznam v tabulce není, využije k nalezení protokol ARP[3] (Address Resolution Protocol). Přepínač přiřadí získanou MAC adresu k příslušnému portu, ze kterého zprávu obdržel. V případě další komunikace s touto stanicí rámec putuje přímo tímto portem [16], [17].

Pro více informací o přepínání na linkové vrstvě doporučuji literaturu [16], nebo pro další informace o počítačových sítích [17].

1.3 Směrování

Směrování (routing) je proces určující cestu paketů od odesílatele k příjemci. Působí na třetí vrstvě ISO/OSI, ale ke svému správnému fungování spolupracuje i s nižšími vrstvami. Směrování je klíčovým rysem internetu, protože řídí veškerý provoz přenášení dat a to za pomocí směrovacích tabulek. Z jejich analýzy se vybírá nejlepší cesta k adresátovi napříč uzly sítě.

[3]ARP protokol přiřazuje IP adresy k fyzickým adresám (MAC) hardwaru. Tento protokol vyšle paket s danou IP adresou na všechny uzly, ten který ji určí jako shodnou s vlastní, pošle zpět fyzickou adresu. Vyhledávání funguje jen v rámci stejné podsítě. ARP se používá u IPv4, pro IPV6 se používá Neighbor Discovery Protocol (NDP) [6].

IP (Internet Protocol) využívá postupu zvaný bitový součin. Ten slouží k určení, zda odesílající i přijímající počítač je na stejné podsíti. Bitový součin je kombinace binárních verzí IP adres obou počítačů s maskou podsítě (prefixem).

Příklad bitového součinu může být následující: IP adresa odesílajícího počítače je 192.168.1.2 s prefixem /24, přijímajícího počítače je IP adresa 192.168.10.4/24. Nejdříve použijeme logickou operaci AND na IP s prefixem odesílajícího počítače:

$$\begin{aligned} 192.168.1.2 &= 11000000.10101000.00000001.00000010 \\ 255.255.255.0 &= 11111111.11111111.11111111.00000000 \\ \text{AND} &= 11000000.10101000.00000001.00000000 \end{aligned}$$

Potom provedeme stejnou operaci u přijímajícího počítače:

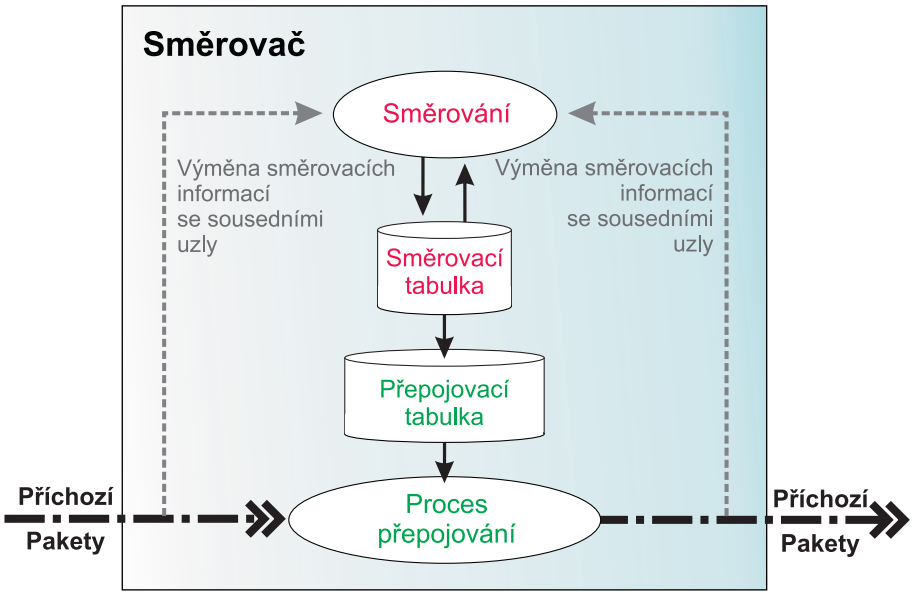
$$\begin{aligned} 192.168.10.3 &= 11000000.10101000.00001010.00000011 \\ 255.255.255.0 &= 11111111.11111111.11111111.00000000 \\ \text{AND} &= 11000000.10101000.00001010.00000000 \end{aligned}$$

Výsledky bitového součinu jsou rozdílné, takže IP „ví“, že tyto 2 počítače jsou umístěny ve 2 různých podsítích. Zpráva je tedy vyslána na směrovač [17].

1.3.1 Směrovač

Typické zařízení určené ke směrování je směrovač (router), v dnešní době jich je ovšem mnohem víc a používají se i L3 přepínače, firewally, servery či jakékoliv počítače vybavené síťovou kartou. Směrovače vzájemně propojují mnoho sítí do jedné větší, nebo rozdělují velkou síť na několik menších [5].

Obecně řečeno, směrovač provádí 2 základní úkony, jimiž jsou směrování a přepojování paketů. Směrování paketů je prováděno staticky nebo dynamicky. Staticky se směrovač řídí konfigurací sestavenou administrátorem a bez jeho dalšího zásahu se směrovací tabulka nemění. Provoz sítě je tímto velmi zrychlen, jelikož směrovač neprovádí složité výpočty skrz výběr nejlepší cesty a navíc nezatěžuje provoz rozesíláním svých směrovacích informací sousedním směrovačům. Naopak u dynamického směrování si směrovač pomocí směrových protokolů vyměňuje mezi sousedními směrovači směrovací informace, tím si každý směrovač vytváří jistou představu o topologii sítě a vypočítá si pro jednotlivé sítě nejlepší cesty, ty si pak ukládá do přepojovací tabulky. Na základě přepojovací tabulky určuje pohyb paketů ze vstupního rozhraní směrovače na příslušný výstupní port směrovače. Rozpoložení těchto funkcí směrovače je zobrazeno v obrázku 1.2. Rychlost tohoto přepojování pak určuje celkový výkon směrovače [13].



Obr. 1.2: Vnitřní architektura směrovače z funkčního pohledu dynamického směrování

Proces přepojování paketů může být rozdělen do dvou podskupin. První podskupina se označuje jako **základní přepojování**, obsahující základní funkce směrovače [13]:

- **Validace záhlaví** – Předtím než bude s příchozím paketem nějakým způsobem nakládáno, zkontroluje se kontrolní součet v záhlaví paketu.
- **Doba života**, neboli TTL (Time To Live) – Směrovač sníží velikost TTL v hlavičce paketu o hodnotu jedna. Pokud TTL dojde k nule, paket se zahodí a adresátovi se pošle zpráva o nedoručení pomocí ICMP (Internet Control Message Protocol), která adresátovi sděluje informaci, že byla překročena doba života.
- **Kontrolní přepočet** – Protože je hodnota TTL upravená, je potřeba aktualizovat kontrolní součet hlavičky.
- **Vyhledání trasy** – Pomocí cílové adresy se vyhledá v tabulce přepojování port, kterým se má paket vyslat. Zde se taky směrovač rozhoduje, jestli paket vyslat jako unicast[4] nebo multicast[5].
- **Fragmentace** – Pokud je maximální přenosová jednotka (MTU) na lince menší než velikost přenášeného paketu, paket se fragmentuje na menší části. Na konci linky se fragmenty seskládají do původního paketu.
- Některé pakety mají zvláštní potřeby pro zpracování, těchto paketů je sice

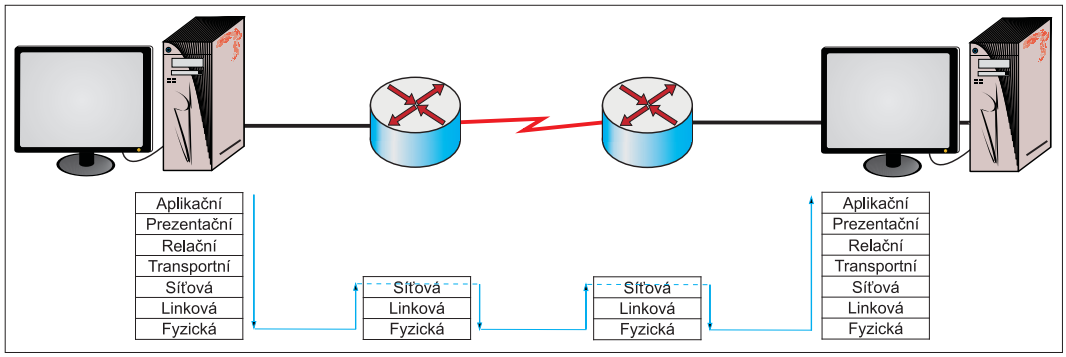
[4]Zaslání paketu jen jednomu cíli.
[5]Zaslání paketu více než jednomu cíli.

málo, ale směrovač je musí umět podporovat.

Druhá podskupina se nazývá **složité přepojování**. Kromě základních funkcí si dnešní svět vynutil další potřeby, jako je vyšší bezpečnost, translace síťových adres, či prioritizace paketů. Tyto aspekty řeší právě podskupina složitého přepojování [13].

1.3.2 Princip směrování

Směrování nastává v okamžiku, kdy je potřeba komunikace do jiné sítě. Využívány jsou k tomu první 3 vrstvy referenčního modelu ISO/OSI, jak jsou přenášena data podle modelu OSI je znázorněno na obrázku 1.3.

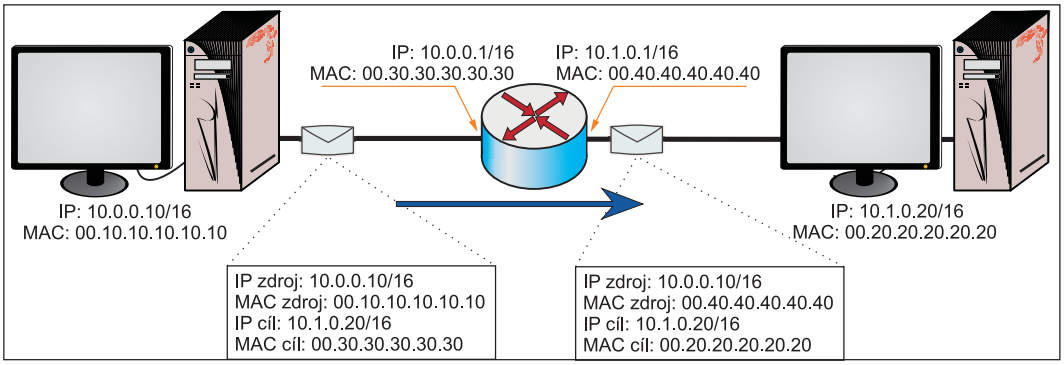


Obr. 1.3: Přenos dat podle referenčního modelu ISO/OSI

Lavina procesů začíná příchodem rámce na směrovač, kde prvně odstraní L2 hlavičku a zakončení, dále pracuje s paketem. Zjišťování dalšího směru paketu na cestě k cílovému adresátovi probíhá tak, že se prvně směrovač rozhoduje, zda je cíl přímo připojený (leží přímo v připojené síti, tedy bude se jednat o nejkonkrétnější síť). Pokud ne, paket se vyšle na další směrovač následujícím způsobem.

Směrovač provede postupně pro každý řádek ve směrovací tabulce bitový součin cílové IP adresy s maskou definované sítě a výsledek se porovná s cílovou IP adresou a maskou z hlavičky přijatého paketu. Může ale vyhovovat více záznamů, v takovém případě se vybere ten, který bude nejvíce vyhovující, viz dále v kapitole 1.3.2. Pro nejvíce vyhovující záznam, směrovač vytvoří záznam do přepojovací tabulky a paket se odešle na odpovídající rozhraní z řádku tabulky. Příští pakety se porovnají s přepojovací tabulkou, a pokud jsou ze stejného streamu, odešlou se stejnou cestou bez opětovného prohledávání směrovací tabulky. Pokud směrovač nenajde žádnou shodu, paket zahodí, nebo odešle na implicitní cestu (jen v případě, že je implicitní cesta nastavena). Ale dříve než směrovač cokoliv odešle, je potřeba paket zpátky zapouzdřit do rámce. Celému procesu odstranění a opětovnému přidání L2 hlavičky se říká přebalování rámce. Proces přebalování je zobrazen na obrázku 1.4. K tomuto

přebalení potřebuje směrovač zjistit fyzické adresy zařízení. Do rámce uvede svoji MAC adresu jako zdrojovou a jako cílovou potřebuje MAC adresu dalšího směrovače. Pokud ji nezná z přepojovací tabulky, směrovač vyšle ARP žádost jako broadcast, zařízení s odpovídající cílovou IP adresou vyšle zpátky jako unicast ARP odpověď. Směrovač si uloží MAC adresu do přepojovací tabulky, kde je dočasně zachována a taky ji přiřadí do rámce, který následně posílá na další směrovač [6], [3], [13].



Obr. 1.4: Přebalování hlavičky rámce

Tento postup se opakuje na každém směrovači směrem k příjemci, dokud zpráva nedojde k cíli.

Směrovací tabulka

Jak už bylo zmiňováno dříve, směrovač orientuje provoz podle směrovací tabulky (příklad tabulky zobrazuje obrázek 1.5). Ta slouží k jistému zmapování síťové topologie a s její pomocí směrovač sestavuje přepojovací tabulku, která má na starost samotné přepojování paketů. Do směrovací tabulky se zapisují informace staticky, dynamicky, nebo automaticky směrovačem, tyto údaje můžeme nazývat „cesty“ nebo „směrovací záznamy“. Staticky znamená, že jsou směrovací údaje přidány ručně administrátorem, dynamicky jsou informace vyměňovány mezi sousedními směrovači na základě použitého směrového protokolu. Automaticky směrovač vytváří záznamy jen u přilehlých sítí a to hned poté, co je nakonfigurováno rozhraní a linka přímo připojená k tomuto směrovači. Směrovací tabulka může obsahovat tyto prvky:

- **Cílová síť** – Je cílová IP adresa podsítě (nikoliv hosta) v intervalu od 0.0.0.0 po 255.255.255.255.
- **Maska** (velikost prefixu) – Záznamy ve směrovací tabulce jsou seřazeny právě podle velikosti masky, která určuje jak hodně je cílová síť konkrétní a to od nejkonkrétnější (nejvíce jedniček) po nejobecnější (nejméně jedniček).
- **Brána** (Gateway) – Zde se zapisuje IP adresa dalšího směrovače (taktéž lze nazývat adresou Next hop).

- **Rozhraní** (Interface) – Výstupní port, nebo adresa výstupního portu směrovače.
- **Metrika** – Značí hodnotu cesty, kde nejnižší metrika je značena nejlepší cestou. Velikost metriky závisí na použitém směrovém protokolu. U Cisco směrovačů se vedle metriky uvádí navíc i administrativní vzdálenost ve tvaru [AD/-metrika].
- **Protokol** – Směrovač může být konfigurován více než jedním protokolem, tato kolonka pak slouží k určení způsobu, jakým byl tento směrovací záznam získán.

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

D    192.168.1.0/24 [90/3014400] via 192.168.12.1, 00:01:39, Serial0/0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
D    192.168.3.0/24 [90/2172416] via 192.168.23.2, 00:01:40, Serial0/0/1
D    192.168.11.0/24 [90/3139840] via 192.168.12.1, 00:01:39, Serial0/0/0
     192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
D       192.168.12.0/24 is a summary, 00:01:39, Null0
C       192.168.12.0/30 is directly connected, Serial0/0/0
D    192.168.13.0/24 [90/41024000] via 192.168.23.2, 00:01:40, Serial0/0/1
                     [90/41024000] via 192.168.12.1, 00:01:39, Serial0/0/0
     192.168.22.0/30 is subnetted, 1 subnets
C       192.168.22.0 is directly connected, Loopback1
     192.168.23.0/24 is variably subnetted, 2 subnets, 2 masks
D       192.168.23.0/24 is a summary, 00:01:48, Null0
C       192.168.23.0/30 is directly connected, Serial0/0/1
D    192.168.33.0/24 [90/2297856] via 192.168.23.2, 00:01:40, Serial0/0/1
S*   0.0.0.0/0 is directly connected, Loopback1
```

Obr. 1.5: Ukázka směrovací tabulky Cisco směrovače

Výběr směrovacích cest

Ve směrovací tabulce může být spousta možných cest k cíli, výběr té nejlepší určují tyto 3 aspekty [1]:

- **délka prefixu**,
- **administrativní vzdálenost** (mezi jednotlivými směrovými protokoly),
- **metrika** (v rámci směrového protokolu).

První prioritou výběru cesty je **délka prefixu**, kde upřednostňujeme cesty s nejdelším prefixem, tedy nejkonkrétnější sítě.

V případě, že je směrovač konfigurován více směrovými protokoly, pro určení nejlepší cesty použije **administrativní vzdálenost**. Ta vyjadřuje kvalitu (důvěryhodnost) daného směrového protokolu. Hodnoty administrativních vzdáleností pro směrovače je možné konfigurovat dle vlastních preferencí, defaultní tabulka těchto hodnot je vyobrazena v tabulce 1.1 [1].

Tab. 1.1: Tabulka hodnot administrativních vzdáleností

Zdroj cesty	**Administrativní vzdálenost**
Přímo připojené rozhraní	0
Statická cesta	1
Součet EIGRP	5
Externí BGP	20
Interní EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIPv1, RIPv2	120
EGP	140
Externí EIGRP	170
Interní BGP	200
Neznámá	255

Dalším důležitým parametrem při výběru je **metrika**. Pokud existuje více cest do stejné sítě zajištěných jedním směrovým protokolem, směrovač na základě velikosti metriky vybere tu nejlepší (nejmenší hodnota značí nejlepší cestu). Každý směrový protokol využívá odlišné výpočty pro stanovení metriky, velikost metriky tedy závisí na použitém směrovém protokolu. Hodnotu metriky stanovuje například: počet přeskoků, šířka pásma, spolehlivost (četnost chybovosti linky), zpoždění a cena (staticky přiřazena administrátorem) [5].

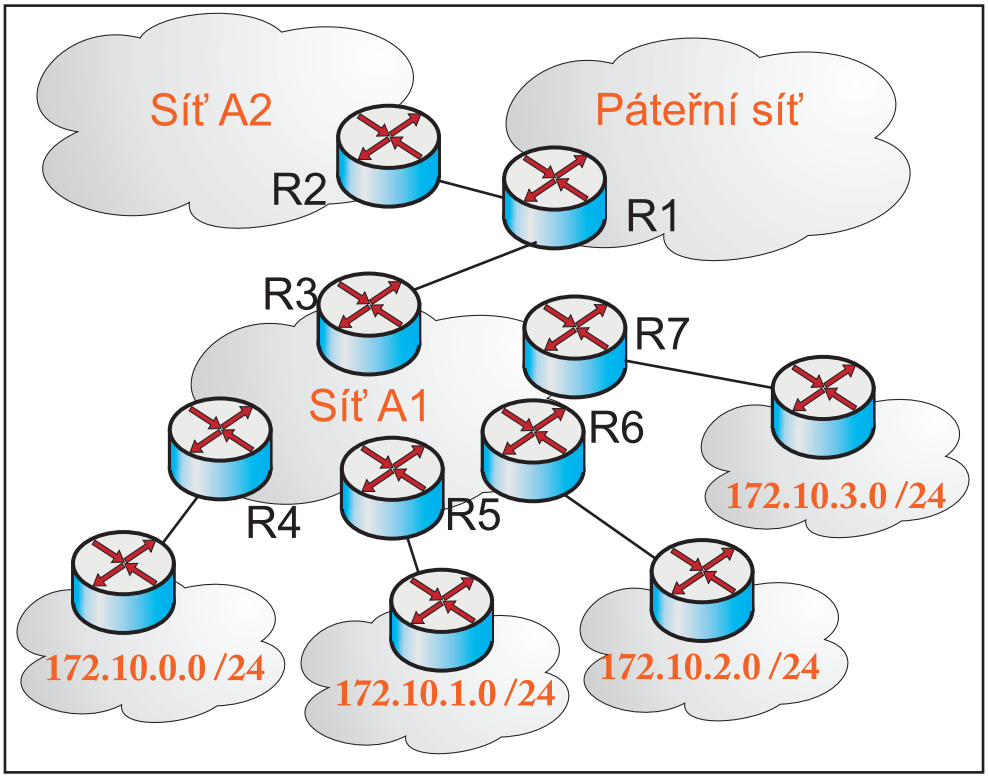
Obecně směrovači od sousedů přichází spousta směrovacích údajů o okolních sítích. Pro jednu vzdálenou síť tak může získat třeba 10 různých cest a zapsat je všechny do směrovací tabulky by bylo zbytečné. V tabulce chceme mít co nejméně údajů, protože při vyhledávání další cesty musí směrovač procházet celou tabulku, to nejen zpomaluje provoz, ale vytěžuje centrální procesorovou jednotku (CPU) směrovače. Směrovací tabulku bez zásahu administrátora primárně zjednodušuje metrika a administrativní vzdálenost, které v jistých případech přijímané cesty ani nezapíše do tabulky. Takový případ může nastat například u protokolu RIP (více v kapitole

1.4.3), kde metrika představuje počet přeskoků k cíli. Směrovači, s tímto dynamickým protokolem, může přicházet spousta směrovacích informací obsahující různé cesty do jedné vzdálené sítě, zapisovat si do směrovací tabulky všechny tyto možné cesty by bylo zbytečně zatěžující, proto si směrovač uloží jen cesty s nejnižším počtem mezilehlých směrovačů. V dalším případě směrovač nezapisuje směrující záznamy, které přišly z neznámého zdroje, jelikož je považuje za vysoce nedůvěryhodné [1].

Sumarizace sítí

Sumarizace, neboli agregace, je proces zápisu více směrovacích informací do jedné. Výhodou tohoto procesu je snížení počtu záznamů ve směrovací tabulce. Jelikož při určování cesty se prochází celá směrovací tabulka, urychlí se tím i proces vyhledávání. Aby bylo možné cesty sumarizovat, musí platit tyto dvě podmínky:

- cílové sítě mohou být sumarizovány do jedné síťové adresy,
- cílové sítě musejí být směrovány skrz stejné rozhraní, či Next-hop.



Obr. 1.6: Příklad sumarizace linek

Sumarizace mohou být konfigurovány manuálně administrátorem, nebo automaticky směrovým protokolem, ovšem v dnešní době se využívá výhradně spíše manuální konfigurace. Pro příklad výpočtu poslouží obrázek 1.6. Směrovač R1 má ve směrovací tabulce celkem 4 záznamy, které mohou být sumarizované. Na agregaci

jsou tedy potřeba 2 bity z prefixu. Výsledná sumarizovaná adresa bude 172.10.0.0/22. Časem může nastat případ, že například síť 172.10.2.0/24 bude chtít změnit poskytovatele a přejde do sítě A2. Sumarizovaná adresa směrovače R1 může stále existovat, ale musí se zároveň přidat do přepojovací tabulky výjimka 172.10.2.0/24 směřující do sítě 2. Jelikož má tato adresa konkrétnější prefix, bude upřednostněna před sumarizovanou adresou [13].

Přepojovací tabulka

Pomocí směrovací tabulky se sestavuje přepojovací tabulka (ukázka viz obr. 1.7), ta se hlavně vztahuje k rychlému přepojování paketů ze vstupního rozhraní na příslušné výstupní rozhraní směrovače. Je optimalizovaná na základě IP prefixu cílové sítě pro rychlé hledání cesty. Přepojovací tabulka obsahuje tyto položky [13]:

- **Cílová síť** a **maska**.
- **Rozhraní** – Výstupní port směrovače určující budoucí směr paketu.
- **MAC adresa** – Fyzická adresa zařízení dalšího skoku.

Cílová IP a prefix	Rozhraní	MAC adresa
10.5.0.0/16	fa0/0	00.10.16.a0.ab.11
172.16.1.0/24	fa0/1	00.30.23.dc.01.d8

Obr. 1.7: Ukázka přepojovací tabulky

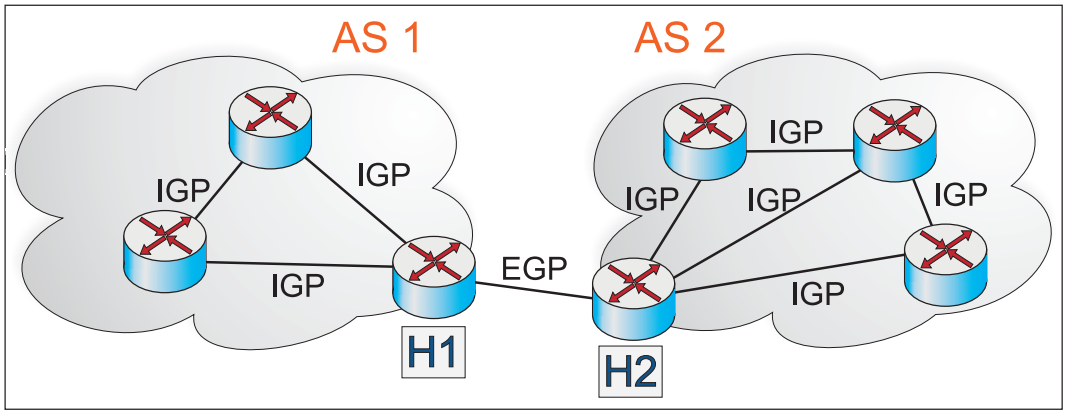
V závislosti na použité technologii směrovače se MAC adresa do přepojovací tabulky nezapisuje. Pro zápis MAC adres se v tomto případě používá navíc další tabulka (tzv. tabulka sousedství), ve které záznam obsahuje opět cílovou IP adresu s prefixem a dále již zmiňovanou MAC adresu zařízení dalšího skoku [3].

1.3.3 Autonomní systémy

Vzhledem k rozsáhlosti dnešního Internetu je nemožné, aby směrovač znal celou síťovou topologii. Navíc v okamžiku výpadku v síti je potřeba aktualizovat směrovací tabulku všech směrovačů a než by aktualizace došla na druhou stranu Internetu, nastaly by mezitím další desítky výpadků a Internet by se stával nestabilním. Z těchto důvodů se zavedlo dělení Internetu na tzv. autonomní systémy (AS) – Autonomous Systems.

AS zahrnuje skupinu sítí a směrovačů pod společnou směrovací politiku a správu. Společná politika zahrnuje hlavně druh použitého směrového protokolu, dále řeší speciální požadavky administrátora, například Load Balancing (rozložení zátěže do

více linek). AS může být dále rozdělen na ještě menší části, zvané oblasti, ty využívá například protokol OSPF (věnováno kapitole 1.4.3).



Obr. 1.8: EGP a IGP v autonomních systémech

Obr. 1.8 zobrazuje rozdělení směrování do dvou úrovní. V rámci komunikace uvnitř autonomních systémů se používají interní směrové protokoly – Interior Gateway Protocols (IGP). Mezi autonomními systémy se naopak využívá externích směrových protokolů - Exterior Gateway Protocols (EGP). Směrovače H1 a H2 se nazývají hraniční směrovače (ASBR u protokolu OSPF), ty se obecně starají o výměnu souhrnných směrovacích informací mezi různými směrovými protokoly. V tomto případě hraniční směrovače propojují autonomní systémy a jsou umístěny mezi EGP a IGP.

Jako IP adresy sítí i autonomní systémy musí být jednoznačně identifikovány pro správné směrování mezi autonomními systémy. Takový identifikátor býval dříve reprezentován šestnácti bity, ale stejně jako docházely volné adresy IPv4, začaly docházet i 16 bitové autonomní systémy. U IPv4 se zavedlo řešení postupného přecházení na IPv6 (128 bitová adresa), u autonomních systémů se problém vyřešil podobně a to s přestupem na 32 bitové ASN (Autonomus System Number) roku 2009 [10].

Autonomní systémy se dále dělí podle počtu dalších připojených systémů [10]:

- **Single-homed AS** – Existuje jen jedno spojení s dalším asynchronním systémem.
- **Netranzitní multi-homed AS** – Je propojen z důvodu redundance k více autonomním systémům.
- **Tranzitní multi-homed AS** – Spojuje více autonomních systémů, mezi kterými si může vyměňovat směrovací informace a data.

1.4 Metody směrování

Směrování je primární vlastností směrovače, ke správné funkci ovšem potřebuje již zmíněnou směrovací tabulku. Záznamy v této tabulce jsou zapisovány pomocí dvou základních metod, které nazýváme statické nebo dynamické směrování.

Samotný směrovač může být umístěn mezi 2 sítěmi, kde každá sít podporuje jiný směrový protokol. Aby směrovač mohl podporovat více směrových protokolů je zapotřebí zajistit mezi těmito protokoly sdílení informací. Tomuto propojení se říká redistribuce. Nejčastěji se redistribuce používá v kombinaci mezi BGP a nějakým dalším dynamickým protokolem.

1.4.1 Statické směrování

Zápis do směrovací tabulky statickým směrováním je spravován ručně pomocí administrátora, kde pro plnou konektivitu musí být statické směrování nakonfigurováno na všech zařízeních, která musí rovněž mít nakonfigurovány cesty do všech vzdálených sítích. Taková manuální konfigurace bývá náročná a ve větších sítích takřka nemožná, proto má statické směrování využití jen v menších a jednodušších sítích.

U statického směrování směrovače mezi sebou nesdílejí směrovací záznamy. Výhodou toho je především bezpečnost, jelikož zde není možné jakkoli odposlouchávat topologii sítě, či pozměnit sdílené směrovací informace. Dále má navíc směrovač velmi snížené zatížení CPU i minimální využití paměti (RAM) a mezi směrovači se šetří šířka pásma, jelikož statické směrování nezatěžuje provoz. Avšak v důsledku nesdílení směrovacích záznamů mezi směrovači je velkou nevýhodou odolnost vůči změnám v síťové infrastruktuře. V případě přidání nové sítě, či výpadku nějaké linky se změny neprojeví ve směrovací tabulce a je opět vyžadován zásah administrátora [15].

Defaultní administrativní vzdálenost pro statické směrování je 1, pokud tato hodnota není změněna, je statické směrování do vzdálené sítě vždy upřednostněno před dynamickým směrováním.

Použití statického směrování má zastoupení hlavně v menších a jednodušších sítích, kde nedochází k častým změnám v topologii. Staticky lze taky nakonfigurovat implicitní cestu (default route), která se používá v případě, pokud neexistuje žádná jiná cesta k cíli. Implicitní cesta má tvar cílové sítě a prefixu 0.0.0.0/0, dále se určuje, kterým přepojovacím rozhraním směrovače, či na jakou IP adresu dalšího skoku se má přenášená informace odeslat. Používá se zpravidla ke směrování do Internetu [1].

1.4.2 Dynamické směrování a jeho algoritmy

Vzhledem k rozsáhlosti většiny dnešních datových sítí je statická konfigurace a údržba směrovačů administrátorem nemožná. Z tohoto důvodu se zavedlo dynamické směrování, kde je zásah administrátora vyžadován nejvíce při prvotní konfiguraci protokolu směrovače, kde administrátor přidává sousední sítě, s kterými si má směrovač vyměňovat směrovací záznamy. Zavedení dynamického směrování do velkých sítí navíc snižuji riziko pro chyby ve vstupech do směrovacích tabulek.

Pomocí směrových protokolů spolu směrovače vzájemně komunikují a automaticky si vytváří i aktualizují informace ve svých směrovacích tabulkách. Ale v důsledku toho je mezi směrovači vyšší provoz (hlavně při startu, protože si směrovače potřebují předat všechny směrovací informace o ostatních linkách), což spotřebovává šířku pásma. Rovněž jsou směrovače více vytěžovány, jak z hlediska paměti, tak i CPU a proto jsou na směrovače s dynamickým směrováním kladeny vyšší nároky než u statického směrování [1].

Avšak dynamické směrování má i spoustu výhod. Když dojde k výpadku v síťové infrastruktuře, směrovač vypočítá a vybere jinou nejlepší cestu vedoucí k cíli (výběr cesty popsán v kapitole 1.3.2. Důležitou roli tu hraje doba konvergence, což je v případě výpadku uzlu v síti čas, potřebný pro konverzi směrového protokolu. Konvergence je opět dosažena v okamžiku, kdy je všem směrovačům v síti ohlášen daný výpadek a tato změna v topologii je zaznamenána do směrovací tabulky u všech směrovačů [1], [4].

Zavedení dynamického směrovacího protokolu se používá výhradně ve velkých či středních sítích, naopak pro malé sítě se nedoporučuje, docházelo by zde k zbytečnému zatěžování spojů [1].

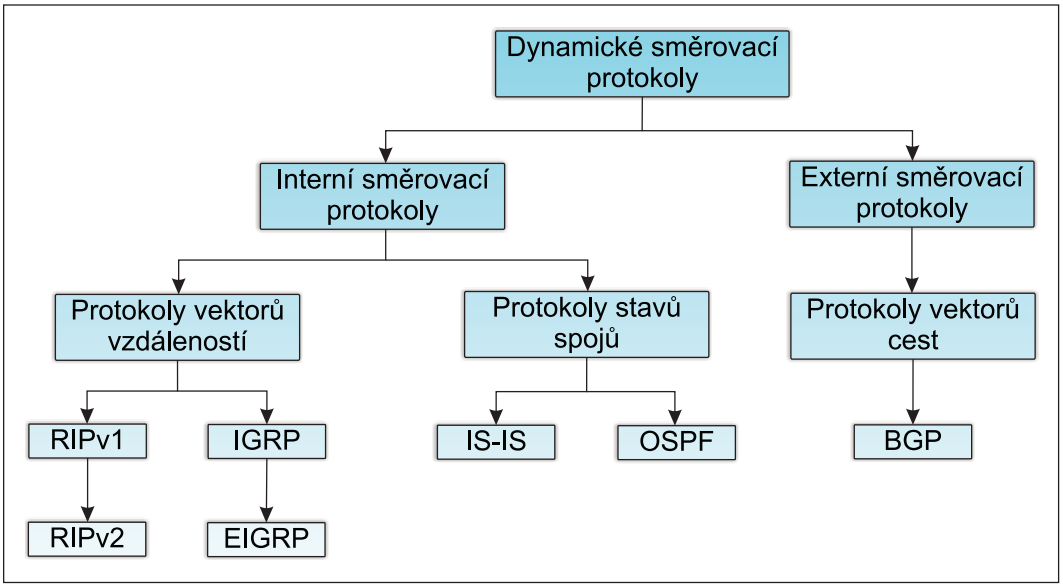
Jak už bylo napsáno dříve, dynamické protokoly se dělí na **interní** (IGP) a **externí** (EGP). Interní směrové protokoly se dále dělí do 2 základních skupin [4]:

- **protokoly vektorů vzdáleností** (DVA),
- **protokoly stavů spojů** (LSA).

Existuje i metoda směrování, která přebírá funkce z obou předešlých skupin. Tato kombinace se někdy označuje jako hybridní protokol, avšak správnější název této metody je **pokročilejší protokol vektorů vzdáleností**. Externí směrové protokoly pracují pod skupinou:

- **protokolů vektorů cest** (PVA),

tyto protokoly jsou taky jistou nadstavbou DVA. PVA je především upraven do podoby, aby rozuměl autonomním systémům. Celé rozdělení stručně zobrazuje obrázek 1.9 [4].



Obr. 1.9: Dělení dynamických směrových protokolů

Algoritmus vektorů vzdáleností

Směrovače používající algoritmus vektorů vzdálenostní (Distance Vector Algorithm) neznají topologii sítě, ale znají rozhraní sousedních uzlů, přes která směrovat datové informace do cílových sítí a vzdálenosti (počty přeskoků) k těmto sítím. Pro určení nejkratší vzdálenosti (cesty) od všech ostatních uzlů je použit Bellman-Fordův algoritmus.

Výměna směrových informací v tomto případě probíhá tak, že směrovače periodicky vysílají obsah své směrovací tabulky všem okolním uzlům a to i v případě, že v síti nenastala změna. Nevýhodou periodického vysílání je hlavně pomalá konvergence, protože při změně v topologii se vzdálený uzel dozví o tomto stavu až při příštím periodickém všesměrovém odeslání směrovací tabulky. Další nevýhodou je při výpadku v síti tvorba nekonečných smyček (po dobu dokud není obnovena konvergence).

Když směrovač obdrží směrovací informaci od souseda, navýší hodnotu vzdálenosti o 1 (kromě přímo připojených uzlů) a porovná ji s informací ve své směrovací tabulce. Pokud obdrženou směrovací informaci nezná, přidá si ji do své směrovací tabulky, pokud je ve směrovací informaci známá síť s menší vzdáleností, směrovací tabulku si aktualizuje. V případě, kdy není cesta do vzdálené sítě delší dobu obdržena od souseda, cesta se ze směrovací tabulky odstraní. Po úpravě své směrovací tabulky, ji směrovač celou odešle všesměrově dalším sousedům. Ovšem odesílání celé tabulky zatěžuje síť a to především u větších sítí. Proto se protokoly vektorů vzdáleností používají výhradně ve středních či malých sítích, kde plní svoji funkci lépe.

Protokoly DVA používají k určení nejlepší cesty metriku, která je vypočtena na základě jednoho nebo více kritérií. Zástupce, který používá jedno kritérium, je RIP (Routing Information Protocol). Tento protokol používá k určení metriky pouze počet přeskoků. Zástupce používající více kritérií je IGRP (Interior Gateway Routing Protocol), který k výpočtu metriky nepoužívá jen počet přeskoků, ale i dobu zpoždění či propustnost linky [1], [4].

Aby se předešlo problémům spojených s DVA (například nekonečné smyčky, nebo počítání do nekonečna) zavedlo se následujících funkcí [2]:

- **Split Horizon** – Když směrovač obdrží špatnou informaci od sousedního směrovače, například při výpadku, směrovač si tuto chybu sice poznamená do své tabulky, ale s příchodem tabulky od sousedního směrovače (ještě neví, že daná síť za prvním směrovačem vypadla) si první směrovač opět přepíše svoji tabulku a pošle ji zpět sousedovi. Takhle se navyšuje vzdálenost do nekonečna. Split Horizon této činnosti zabraňuje tak, že směrovač neposílá směrovací informace o síti na rozhraní, přes které přišly.
- **Poison Reverse** – Obdobný způsob jako Split Horizon, avšak zde se u rozhraní, odkud přišly směrovací informace, nastavuje maximální doba života (TTL).
- **Triggered Update** – Po výpadku, náběhu přilehlé linky, nebo příchodu Update informace od souseda se nečeká periodická doba, ale směrovací informace se ihned odešlou dále ostatním sousedům. To značně zrychluje dobu konvergence.
- **Hold Time** – V případě, kdy směrovač obdrží update informaci o výpadku sítě, nebere určitou dobu v úvahu žádné další informace o dané síti. Mohlo by se totiž jednat o klamavé informace od směrovačů, které se o výpadku zatím nedozvěděly.

Algoritmus stavů spojů

Algoritmus stavů spojů (Link State Algorithm) byl vytvořen, aby překonal nedokonalosti DVA, jako je tvorba nekonečných smyček, pomalá konvergence, či posílání celých tabulek, čímž se značně zatěžují právě velké sítě. Protokoly LSA si udržují 3 tabulky [1]:

- **tabulku sousedů**, která obsahuje všechny připojené sousedy a rozhraní sousedů, tato tabulka je vytvořena pomocí tzv. Hello paketů,
- **tabulka (databáze) síťové topologie**, obsahuje mapu linek celé topologie v rámci oblasti a taky stav těchto linek,
- **tabulka nejkratších cest**, obsahuje nejlepší cesty do vzdálených uzlů, pro všechny směrovače v oblasti je stejná.

Směrovače, používající protokoly stavů spojů, znají celou topologii sítě a udržují databázi stavů spojů. Rychle reagují na změny v síti. Každý směrovač hlídá dostupnost svých sousedů, ke kterým je připojen, a v případě nějaké změny zasílá ihned jen danou informaci (pomocí LSP – Link State Packet) hierarchicky všem uzlům v síti. Aktualizace se zasílají pouze při změně, avšak minimálně jednou za 30 minut. Jelikož se posílá informace jen o daném stavu, LSA protokoly spotřebovávají velkou šířku pásma jen při startu, kdy si všechny směrovače musejí vyměnit LSP, následná spotřeba šířky pásma je mnohem menší než u DVA a není nijak závislá na velikosti sítě, jelikož vysílaná aktualizace bude vždy stejné velikosti. Avšak je zde kladen vyšší nárok na CPU a RAM samotného směrovače.

Protokoly LSA jsou vhodné pro velmi rozsáhlé sítě, pro lepší efektivitu se navíc rozdělují do menších oblastí, aby směrovač nemusel rozumět tak velkému množství směrovacích informací (na hranicích oblasti se cesty sumarizují).

Směrovače ve stejné oblasti si udržují stejnou směrovací tabulku, a konvergence je zde velmi rychlá, jelikož je na změnu ihned reagováno. Díky tomu jsou LSA protokoly imunitní vůči nekonečným smyčkám.

Tabulka nejkratších cest se stanovuje podle nejmenší metriky, na kterou má vliv například šířka pásma, zpoždění, bezpečnost, atd.... K výpočtu nejkratší cesty slouží Dijkstrův algoritmus SPF (Shortest Path First), to probíhá tak, že se sestavuje strom nejkratších cest od zdroje ke všem uzlům. Odesílání zpráv začíná u kořene a opakuje se do té doby, dokud nedostane informace od všech uzlů v sítí. Ze všech uzlů v síti se pak vybere uzel, ke kterému vede nejkratší cesta od kořene [1].

Typickými zastupiteli LSA protokolů jsou:

- **OSPF** (Open Shortest Path First),
- **IS-IS** (Intermediate System-to-Intermediate System).

Pokročilý algoritmus vektorů vzdáleností

Tato metoda, s anglickým názvem Advanced Distance Vector Algorithm, obsahuje v značné míře prvky LSA, avšak jedná se o pokročilou verzi DVA. Obecně se dá říct, že si uzly nevyměňují celé směrovací tabulky, ale přeposílají dál jen informaci o změně (stejně jako u LSA využívá k určení konektivity Hello pakety), díky tomu spotřebovává nízké síťové prostředky a disponuje velmi rychlou konvergencí. Stejně jako DVA nezná celou topologii, ale jen rozhraní, přes která směrovat.

Největším rozdílem oproti DVA a LSA je, že nepoužívá Bellman-Fordův ani Dijkstrův algoritmus, ale difuzní algoritmus aktualizace (DUAL), který využívá vzdálenostní informace (metriky) k určení efektivní cesty bez smyček. Nejkratší součet metrik cesty udává primární trasu. Navíc je počítaná i trasa, která v případě výpadku primární trasy převezme provoz (označována jako následník). Ta se

určuje, když je součet metrik druhý nejkratší, ovšem musí splňovat podmínku, že metrika sousední linky, přes kterou má být směrován provoz, musí být menší než součet metrik primární cesty. Dále DUAL využívá aktivní a pasivní status. Těchto statusů se využívá v případě změny metriky, či výpadku přilehlé linky. Směrovač si změní pasivní status za aktivní a ostatním sousedům pošle informace o dané změně. V případě výpadku a v následném důsledku nějaký směrovač ztratí svou jedinou cestu napříč sítí, tento směrovač si změní stav za aktivní a pošle sousedům dotaz, kterým hledá novou cestu. Směrovač, který zná novou cestu, pošle zpátky odpověď, ve které jsou obsaženy nové směrovací informace a směrovač, který měl dříve status aktivní, si jej opět změní na pasivní [13].

Tato metoda patří mezi nejnovější a zatím má jediného zástupce, kterým je EIGRP protokol, vynalezený firmou Cisco.

Algoritmus vektorů cest

Protokoly založené na algoritmu vektorů cest (Path Vector Algorithm) nepracují s grafem propojených směrovačů či sítí (jako u DVA), ale s grafem propojení autonomních systémů. K správnému směrování využívá tzv. vektor cest (Path Vector), který obsahuje posloupnost čísel autonomních systémů, napříč kterými má informace putovat k cílové síti. Čím více jsou cílové sítě vzdáleny, neboli přes čím více autonomních systémů musí zpráva projít, aby se dostala do cílové vzdálené sítě, tím je vektor cest delší. Při určování vektoru cest je vybírán ten nejkratší, tedy ten, co prochází nejmenším počtem autonomních systémů. Jelikož i u směrování napříč autonomními systémy mohou vznikat nekonečné smyčky, může se číslo autonomního systému vyskytovat ve vektoru cest pouze jednou. Tak se nikdy nedostane znovu do autonomního systému, kde už byl v minulosti.

Typickým zastupitelem používaným v dnešní době je BGP (Border Gateway Protocol) [10].

1.4.3 Protokoly dynamického směrování

Tyto směrové protokoly dynamicky zapisují a udržují cesty ve směrovací tabulce. Jsou zde stručně popsány protokoly RIP a OSPF.

Směrový protokol RIP

Tento protokol patří mezi první protokoly dynamického směrování a spadá pod třídu DVA protokolů, ze které vyplývají i jeho vlastnosti. K výpočtu metriky využívá pouze počtu přeskoků k cíli, kde dovoluje jen 15 přeskoků. Doba života paketu je totiž stanovena na 16 (na 16. směrovači hodnota TTL klesne na 0). Síť vzdálená 16

a více přeskoků přes směrovače se označuje za nedosažitelnou (metrika je nastavena na nekonečno) a síť s hodnotou přeskoku 1 indikuje jako přímo připojenou síť.

RIP ke své funkčnosti používá 4 časovače (s defaultními hodnotami):

- **Update Timer** (30 sekund) – sděluje, jak často má být sousedním směrovačům posílána celá svoje směrovací tabulka.
- **Invalid Timer** (180 sekund) – doba, do kdy je řádek ve směrovací tabulce platný, pokud by od dané cesty nepřicházely žádné aktualizace. Po jejím vypršení je cestě přiřazena metrika 16, pokud tedy během odpočítávání směrovač neobdržel aktualizaci, v tom případě by byl časovač anulován. Po vypršení doby je směrovač ve stavu hold-down.
- **Hold-down Timer** (180 sekund) – Doba, po kterou je cesta přidržena. Směrovač v tomto časovém rozmezí rovněž nepřijímá žádné směrovací aktualizace.
- **Flush Timer** (240 sekund) – Běží současně s časovačem Invalid Timer, tudíž po 60 sekundách, kdy byla cesta označena za neplatnou, je cesta vymazána z tabulky.

Tyto hodnoty časovačů musí být stejné na všech směrovačích v RIP síti, jinak je způsobena nestabilita sítě.

V případě, kdy k cíli vede více cest se stejnou metrikou, je zátěž mezi těmito linkami (defaultně až 4 linkami) vyvažována.

RIP aktuálně nabízí 3 verze: starší je označována jako RIPv1, novější RIPv2 a dále pro podporu IPv6 síťování byl vytvořen RIPng.

RIPv1 podporuje pouze třídní adresování (classful) A, B, C (s prefixy /8, /16, /24). Při konfiguraci tedy není potřeba zadávat masku sítě, avšak tato metoda je velmi nevýhodná z hlediska špatného rozdělení počtu podsítí (třída A má k dispozici málo podsítí a třída C má naopak hodně podsítí) a neefektivního sumarizování sítí. Tento protokol posílá update informace na všesměrovou adresu 255.255.255.255. Dnes se tato verze prakticky nepoužívá.

Velkým rozdílem oproti RIPv1 má **RIPv2** podporu třídního adresování (classless), v sítích RIP tak můžeme používat proměnné délky masky podsítě (VLSM) a díky tomu sítě i efektivně sumarizovat, směrovač má pak uložené daleko méně adres než u RIPv1. Dále tento protokol podporuje autentizaci směrovacích informací s šifrováním MD5 a odesílá je prostřednictvím multicastové adresy 224.0.0.9. RIPv2 umí odesílat směrovací aktualizace jak ve verzi 2, tak i ve verzi 1, je tedy kompatibilní s RIPv1, avšak přijímá pouze směrovací aktualizace ve verzi 2.

RIPng se svými vlastnosti podobá verzi 2, avšak ke sdílení směrovacích informací používá v hlavičce paketu pro IP adresy velikost 128 bitů místo 32 bitů. A dále podporuje, místo autentizace MD5, tzv. IPsec.

Díky své jednoduchosti má svou podporu na široké škále zařízení, avšak jeho implementace má v dnešní době zastoupení jen v menších, či středních sítích v podobě RIPv2 [1].

Směrový protokol OSPF

Tento protokol spadá pod rodinu LSA protokolů a jeho vznik se připisuje k období mezi roky 1988 a 1991. Jako metriku využívá „cenu spoje“, která je ovlivňována šířkou pásma. K jejímu výpočtu je použit následující vzorec:

$$cena\ spoje = \frac{100\ \text{Mb}}{\text{šířka pásma v Mb}},$$

ten ovšem platí při defaultním nastavení směrovačů. Pokud například budeme chtít rozlišit ceny spojů s vyšší rychlostí než 100 Mb (fast ethernet), můžeme změnit nastavenou referenční šířku pásma od které se odpočítává metrika ze 100 Mb na 1000 Mb. Metrika tak nebude přenosové šířce pásma nad 100 Mb přiřazována vždy na 1, ale může se dále rozlišovat od 1 do 10. Metrika cesty je součtem všech cen spojů na cestě k cíli a jako nejlepší cesta je označena ta s nejnižší hodnotou. Počet přeskoků k cíli metriku neovlivňuje, navíc u OSPF není počet přeskoků nijak limitován. Cenu spoje lze taky přepsat manuálně, kde pak můžeme upřednostnit pomalejší spoj nad rychlejším a tak zbytečně nezatěžovat hlavní spoj v případě nízkých požadavcích na přenos. K určení nejkratší cesty se používá Dijkstrův algoritmus [1], [11].

OSPF směrovače si mezi sebou v rámci stejné oblasti udržují tzv. **vztah sousednosti** vyměňováním Hello paketů (na multicastové adrese 224.0.0.5 a 224.0.0.6). Až po sestavení této sousednosti si směrovače mohou přeposílat směrovací informace. Hello pakety jsou defaultně odesílané v pravidelných intervalech [1]:

- **Hello interval** – Slouží k udržení sousedství, posílá se každých 10 s u typů Broadcast a Point-to-Point a každých 30 s u typů Non-broadcast a Point-to-Multipoint.
- **Dead interval** – Pokud po tuto dobu směrovač nepřijme žádný Hello packet od sousedního směrovače, dojde k zániku sousedství, každých 40 s (Broadcast a Point-to-Point) a každých 120 s (Non-broadcast a Point-to-Multipoint). Zpravidla 4 násobek Hello intervalu.

Dále je každý OSPF směrovač identifikovaný pomocí „Router ID“, tento identifikátor je unikátní a směrovači je přidělen jedním z těchto tří způsobů [1]:

1. Router ID je přiděleno manuálně,
2. v případě, kdy není určeno manuálně, jako Router ID se směrovači přidělí nejvyšší IP adresa loopback rozhraní,

3. když na směrovači neexistuje žádný loopback, jako Router ID se mu přidělí nejvyšší IP adresa nakonfigurovaného fyzického rozhraní (toto určení se provede vždy jen při zapnutí směrovače, pozdější změny IP adres na volbu Router ID nemá vliv).

Vztah sousednosti je mezi směrovači navázán v okamžiku, kdy směrovač nalezne v přijatém Hello paketu své Router ID (v poli Neighbor). Dále pro navázání vztahu musí oba směrovače mít stejný Hello a Dead interval, patřit do stejné oblasti a do stejného typu oblasti [11].

Po navázání vztahu sousednosti je dalším krokem výměna topologických informací (jinak řečeno Link State Advertisements – LSAs) mezi sousedními směrovači, což má za cíl dosáhnout shodné tabulky sousedů u všech směrovačů v dané oblasti. Na začátku si směrovače vzájemně pošlou Database Description Packet (DBD paket), v něm je vygenerováno náhodné sekvenční číslo SEQ, od kterého se inkrementuje další komunikace a taky se skrz něj určuje, kdo bude Master a kdo Slave v dané komunikační relaci. Master bude směrovač s vyšším Router ID. Dále Master a poté i Slave směrovač odešlou další DBD pakety, obsahující jejich topologické informace. Po obdržení směrovače porovnají obsah DBD paketu se svoji topologickou databází a v případě chybějících či zastaralých záznamů, zažádají o příslušné LSAs položky pomocí paketu Link State Request (LSR). Soused na něj odpoví paketem Link State Update (LSU), který bude obsahovat žádané LSAs položky. Příchod LSU je potvrzován paketem Link State Acknowledgement (LSAck), kde se již posílá jen LSAs hlavička. Navázání vztahu sousednosti a výměnu směrovacích informací shrnutě popisuje následujících 7 stavů [1], [11]:

- **Down** – V tuto chvíli směrovače začínají komunikaci, doposud od sousedního směrovače nebyl obdržen Hello paket.
- **Init** – Směrovač obdržel Hello paket, ale obousměrná komunikace ještě není zřízena.
- **Two-Way** – Obousměrná komunikace je zřízena a v přijatém Hello paketu je v poli Neighbor Router ID vlastního směrovače. V této fázi se dále určuje, v případě kdy je síť typu Multi Access, DR a BDR směrovač.
- **Exstart** – Stanovení Master a Slave směrovače, kde pak Master zahajuje komunikaci.
- **Exchange** – Směrovače si vymění DBD pakety s hlavičkou LSAs.
- **Loading** – Výměna LSR, LSU a LSAck paketů, na jejich základě si směrovače vyměňují své LSAs.
- **Full** – V tuto chvíli jsou směrovače plně synchronizované. Tato fáze, vzhledem k „postavení“ sousedního směrovače, může být dále značena jako:
  - full/DR – značí sousední směrovač v roli Designated Router,

  - full/BDR – značí sousední směrovač v roli Backup Designated Router,
  - full/BROther – sousední směrovač není v roli DR, ani BDR.

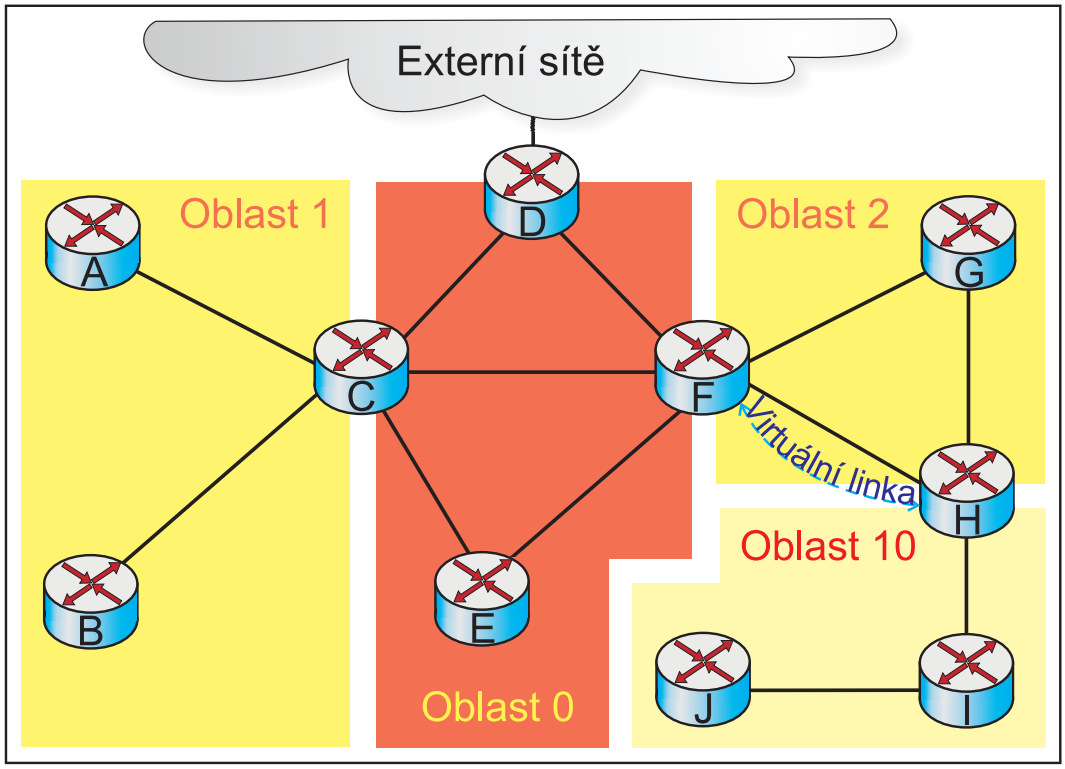
Sítě OSPF lze dělit do 4 typů [1], [11]:

- **Broadcast Multi Access** – Označuje topologii, kde je směrovač propojen s více než 2 počítači, či směrovači na stejné síti (například spojení za pomocí přepínače, nebo rozbočovače). V této síti pak dochází k vyššímu režijnímu provozu a k vyššímu zatížení procesoru směrovačů. Počet odeslaných LSP u takové sítě je dáno vzorcem $n(n-1)/2$, kde n značí počet směrovačů v síti. Proto se v této síti volí pověřený směrovač DR, se kterým ostatní směrovače navazují vztah sousednosti. V případě jeho výpadku se v síti volí i záložní pověřený směrovač BDR. Tyto směrovače se vybírají na základě nejvyšší OSPF priority (8 bitové číslo) DR a s nejbližší nižší BDR. OSPF priorita je defaultně nastavena na hodnotu 1. V případě remízy se DR a BDR vybírají podle nejvyššího Router ID. Směrovač s nastavenou prioritou 0 se nemůže stát DR ani BDR. OSPF komunikace na této síti probíhá pomocí multicastových paketů používající multicastové adresy 224.0.0.5 (přijímají všechny OSPF směrovače) a 224.0.0.6 (přijímá pouze DR a BDR).
- **Point to Point** – Označuje topologii 2 směrovačů v přímém spojení. Nevolí se zde DR/BDR a pro komunikaci se používá adresa 224.0.0.5.
- **Point to Multipoint** – Označuje topologii, kde 1 rozhraní je připojeno k více směrovačům (taky se dá chápat jako série Point-to-Point linek). DR/BDR se nevolí.
- **Non-broadcast Multi Access Network** (NBMA síť) – označuje topologii s možností propojit více než 2 směrovače avšak bez schopnosti posílat broadcasty. Volí se DR a BDR a komunikace probíhá pomocí unicastů, jelikož OSPF sousedi musejí být manuálně specifikováni.

OSPF byl navrhnut především pro směrování ve velkých sítích, jelikož využívá tzv. hierarchického směrování, kdy je autonomní systém rozdělen na menší oblasti. Směrovače pak nemusejí znát všechny sítě v AS, ale pouze sítě ve své oblasti. Další výhodou tohoto rozdělení je, že změna topologie/sítě se projeví pouze v dané oblasti a ne v ostatních, jelikož jsou adresy sítí ostatních oblastí sumarizovány hraničním směrovačem. Výsledkem takového rozdělení je nižší provoz mezi směrovači a taky snížení zatížení CPU směrovačů. Dle své pozice v oblastech se rozlišují 4 typy směrovačů, které jsou pro lepší orientaci znázorněny obrázkem 1.10 (jejich určení je značeno v hranatých závorkách) [1]:

- **Interní směrovač** (Internal Router) – všechny jeho rozhraní směrují v rámci stejné oblasti. [směrovače A, B, G, I, J]

- **Hraniční směrovač** (Area Border Router) – ABR je umístěn na pomezí minimálně dvou různých oblastí. [směrovače C, F, H]
- **Páteřní směrovač** (Backbone Router) – je umístěn mezi více oblastmi, kde jedna z nich je páteřní. [směrovač E]
- **Autonomní hraniční směrovač** (Autonomous System Border Router) – ASBR je směrovač, který komunikuje s ostatními AS s pomocí BGP protokolu. Nejčastěji je umístěn v páteřní oblasti a redistribuuje cesty pro BGP. Jako ASBR jsou označovány i všechny další směrovače, které jakkoli redistribuují cesty z jiných směrovacích protokolů. [směrovač D]



Obr. 1.10: Rozdělení OSPF do oblastí, typy OSPF směrovačů, virtuální linka

Oblast je značena 32 bitovým číslem, kde oblast 0 značí již zmíněnou páteřní oblast. Ta má za hlavní úkol propojit všechny ostatní oblasti (pomocí směrovačů ABR) a mezi nimi či ven z AS směrovat provoz. Pokud oblast není přímo připojena k oblasti 0, jako je tomu například na obrázku 1.10, kde je oblast 10 připojena k oblasti 2, je možné přes tuto oblast zřídit virtuální linku, která propojí hraniční směrovače oblasti 10 s oblastí 0. Tato metoda se ovšem moc nedoporučuje, už z důvodu vyššího vytížení spoje mezi F a H směrovačem. Vzhledem k rozdělení do menších oblastí je tento protokol velice vhodný do velkých sítí [1], [11].

OSPF oblasti lze dále dělit podle procházejícího provozu na **stub** a **tranzitní**

oblasti. Na již výše zmiňovaném obrázku 1.10 jsou typické stub oblasti označené 1 a 10 a jsou to tedy oblasti, kterými neprochází provoz do dalších oblastí. Tranzitní je vždy oblast 0 a v tomto případě i navíc oblast 2, přes tyto oblasti naopak provoz dále prochází k dalším oblastem. Dále existují ještě další 2 typy oblastí a to **Totally Stubby Area**, ze které existuje jen jediná cesta z oblasti a směrování v tomto případě je jednoduše vyřešeno implicitní cestou. Další typ je **NSSA**, což je stub oblast, která disponuje ASBR směrovačem redistribuujíc jiný směrovací protokol. Jelikož ASBR nemůže být ve stub oblasti existuje tato výjimka, která to umožňuje [11].

Aby OSPF protokol zajistil v rámci dané oblasti všude stejné topologické databáze, využívá tzv. LSAs (Link State Advertisements), což je datová struktura v například DBD, LSU a LSAck paketech. Může obsahovat informace o celé oblasti, nebo taky pouze stav rozhraní směrovače s jeho metrikou. LSAs se skládají z hlavičky a datové části. Hlavička obsahuje informace, které určují: typ LSAs, identifikaci původního paketu (IP směrovače), specifikaci části popisující LSAs databáze, stáří paketu a sekvenční číslo paketu. Datová část je závislá na typu LSAs paketu. Nejvíce se vyskytují následující typy LSAs [1], [11]:

- (Typ 1) **Router LSA** – Je generován všemi směrovači. Směrovač touto cestou informuje ostatní v jeho oblasti o stavu svých linek a dále i o ceně těchto linek.
- (Typ 2) **Network LSA** – Je generován pověřenými směrovači (DR) a obsahuje list všech k němu připojených směrovačů, odesílá se taktéž v rámci své oblasti.
- (Typ 3) **Network Summary LSA** – Je generován ABR, obsahuje sumarizované záznamy všech vzdálených sítí a odesílá je do přilehlých oblastí, kde z oblasti 0 se dále posílá na ostatní oblasti.
- (Typ 4) **ASBR Summary LSA** – Je taky generován ABR, obsahuje záznam o cestě k ASBR a odesílá jej do přilehlých oblastí (ne do oblasti 0). Interní směrovače díky tomu budou znát cestu ven z AS.
- (Typ 5) **AS External LSA** – Je generován ASBR, obsahuje externí cesty redistribuované z protokolu BGP a odesílá je do všech oblastí, kromě stub oblastí. Místo velkého množství směrovacích záznamů, může být odeslána do oblastí jen implicitní cestou vedoucí k tomuto směrovači ven z autonomního systému.
- (Typ 7) **NSSA External LSA** – Je taky generován ASBR, ovšem pouze v oblasti typu NSSA, odesílá se jen v rámci dané oblasti a u ABR je konvergován na již známý LSA typ 5.

OSPF dále podporuje VLSM a v závislosti na konfiguraci směrovače i vyvažování zátěže tzv. Load Balancing. **OSPFv2** je druhá verze OSPF a umožňuje navíc

i autentizace pomocí MD5. **OSPFv3** je třetí verzí OSPF a je obdobou OSPFv2, hlavním rozdílem je používání IPv6 namísto IPv4, dále se lehce odlišuje ve stylu konfigurace.

2 NÁVRH LABORATORNÍ ÚLOHY

Tato kapitola se zabývá hrubým návrhem laboratorní úlohy pro předmět Architektura sítí.

2.1 Požadavky

Ohledně směrování by si studenti měli prohloubit své dosavadní teoretické znalosti a především i praktické zkušenosti. Během realizace budou na úlohu kladeny následující požadavky:

- Úloha by měla studenty seznámit:
  - s implementací statického směrování do menší síťové topologie,
  - s primárními protokoly z rodiny vektorů vzdáleností (RIP) a stavů spojů (OSPF) a jejich odlišnostmi, výhodami i nevýhodami,
  - s faktorem výběru nejlepších směrovacích cest.
  - s konvergencí jednotlivých řešení v případě změny v síti.
- Čas na zpracování úlohy by měl být v rozmezí od 70 do 90 minut.

2.2 Použité prostředky

2.2.1 Linksys Wireless-G Broadboard Router – WRT54GL v1.1

Pro sestavení síťové topologie je k dispozici 6 směrovačů WRT54GL verze 1.1 od firmy Cisco. Tento poměrně levnější směrovač používá ke své činnosti 16 MB RAM, 4 MB flash paměť (z tohoto důvodu musí být obraz firmwaru menší než 3866624 B) a BCM5352 procesor taktovaný na 200 MHz.

Na obrázku 2.1 je zobrazena přední část směrovače, ta signalizuje pomocí 8 LED diod stav směrovače, zda:

- je pod napájením,
- je aktivní DMZ (funkce firewallu, která odděluje LAN a WAN síť),
- je aktivní WLAN (bezdrátová LAN),
- jsou aktivní 4 LAN porty, či WAN port (blikání indikuje síťový provoz).

Obr. 2.1: WRT54GL směrovač z předního pohledu

Na obrázku 2.2 je zobrazena zadní část směrovače. Po stranách jsou umístěny konektory pro připojení wifi antének, wifi ovšem v této úloze nebudeme využívat, takže jsou odpojeny. Dále je k vidění červené tlačítko reset a konektor pro napájení. Nejdůležitější částí jsou zde $4 + 1$ porty RJ-45, které mají podporu fast ethernetu (10/100 Mb/s).

Obr. 2.2: WRT54GL směrovač ze zadního pohledu

Tento směrovač nemá ve svém originálním firmwaru k dispozici vstup do příkazového řádku. K jeho konfiguraci se využívá webového rozhraní, které má nízkou podporu směrových protokolů a možnosti jejich konfigurace. Pro účel této práce je použit open-source operační systém OpenWrt (verze 10.03). Využití WRT54GL

směrovače je zaměřeno převážně jen pro domácí účely. OpenWrt tyto možnosti konfigurace určené prodejcem umožňuje obejít a směrovač bude tak moci používat funkcí, které v originálu nejsou dostupné. Avšak tento základní firmware nepodporuje žádné dynamicky směrovatelné metody. Tento problém je vyřešen směrovacím softwarem Quagga založeným na směrovacím softwarovém balíčku Zebra, který nabízí podporu protokolů RIP, OSPF, IS-IS a BGP.

Ke své funkčnosti Quagga využívá tzv. daemony (programy, které jsou dlouhodobě spuštěny při startu zařízení bez přímého kontaktu s uživatelem), které musí být aktivovány v závislosti na směrových protokolech, které chceme na směrovači nastavovat. Tyto daemony se nazývají:

- **zebra** – rozhraní deklarace a statického směrování,
- **bgpd** – BGP směrový protokol,
- **ospfd** – OSPF směrový protokol,
- **ospf6d** – OSPF IPv6 směrový protokol,
- **ripd** – RIP(v2) směrový protokol,
- **ripngd** – RIP IPv6 směrový protokol.

Každý z těchto daemonů má svůj konfigurační soubor a terminálové rozhraní, která mají své vlastní porty. Do těchto terminálových rozhraních se pak přistupuje pomocí protokolu *telnet* a jejich portů:

- **zebra** – 2601,
- **ripd** – 2602,
- **ripngd** – 2603,
- **ospfd** – 2604,
- **bgpd** – 2605,
- **ospf6d** – 2606,

Terminálové rozhraní, pak umožňuje používání velmi podobných příkazů pro konfiguraci daného směrového protokolu, jako je tomu u směrovačů CISCO [7], [12].

Pro účely laboratorní úlohy jsou k systému OpenWRT nainstalovány tyto balíčky:

- quagga - 0.98.6-5
- quagga-libospf - 0.98.6-5
- quagga-libzebra - 0.98.6-5
- quagga-ospfd - 0.98.6-5
- quagga-ripd - 0.98.6-5
- snmpd - 5.4.2.1-2

2.2.2 Počítač s virtuálním prostředím VirtualBox nebo VMware

Laboratorní úloha bude spravována pomocí dvou počítačů, jelikož je určena pro dva studenty. Oba studenti se tak budou moci podílet na postupu úlohy zároveň.

2.2.3 Síťový monitoring PRTG

PRTG je moderní monitorující software sloužící k sledování zařízení celé Naší síťové infrastruktury, pomocí tzv. senzorů. Senzor je popisován jako součást programu, pomocí kterého je monitorováno zařízení. PRTG nabízí více jak 200 druhů různých senzorů, pomocí kterých je sledována například vytíženost CPU, přenos dat, propustnost, odezva, ztrátovost paketů atd...

PRTG poskytuje licenci zdarma s omezením na použití 100 senzorů zdarma, bez ohledu na rozsáhlost sítě. Což je pro účely laboratorní úlohy velmi dostačující.

2.3 Struktura laboratorní úlohy

Laboratorní úloha by měla být navržena tak, aby pro studenty obsahovala informace v co největší míře a dokázala je seznámit s danou problematikou. Ovšem vše musí být zohledněno dle časové náročnosti, v rámci které se může úloha pozměnit. Každé části musí být věnována doba, při které je zvládnut nejen postup úkolu, ale i teoretické pochopení problematiky.

Osnova úlohy by mohla mít následující strukturu:

(I) **Teoretický úvod:**
   (a) Úvod do problematiky směrování,
   (b) statické směrování,
   (c) dynamické směrování - RIP a OSPF.

1. **Implementace statického směrování.**
2. **Implementace dynamického protokolu RIP.**
3. **Implementace dynamického protokolu OSPF.**

Teoretický úvod bude studenty stručně seznamovat s problematikou směrování a s faktory pro určení nejlepší cesty. V nadcházejícím bodu bude popsán souhrn protokolů RIP a OSPF.

Následuje vlastní zadání úlohy, které je rozděleno do 3 praktických částí, kde každá zastupuje jinou oblast problematiky. V první části budou studenti vyplňovat tabulky směrovačů staticky s doprovodem zadání, které rovněž bude studenty seznamovat s výhodami i nevýhodami daného řešení směrování.

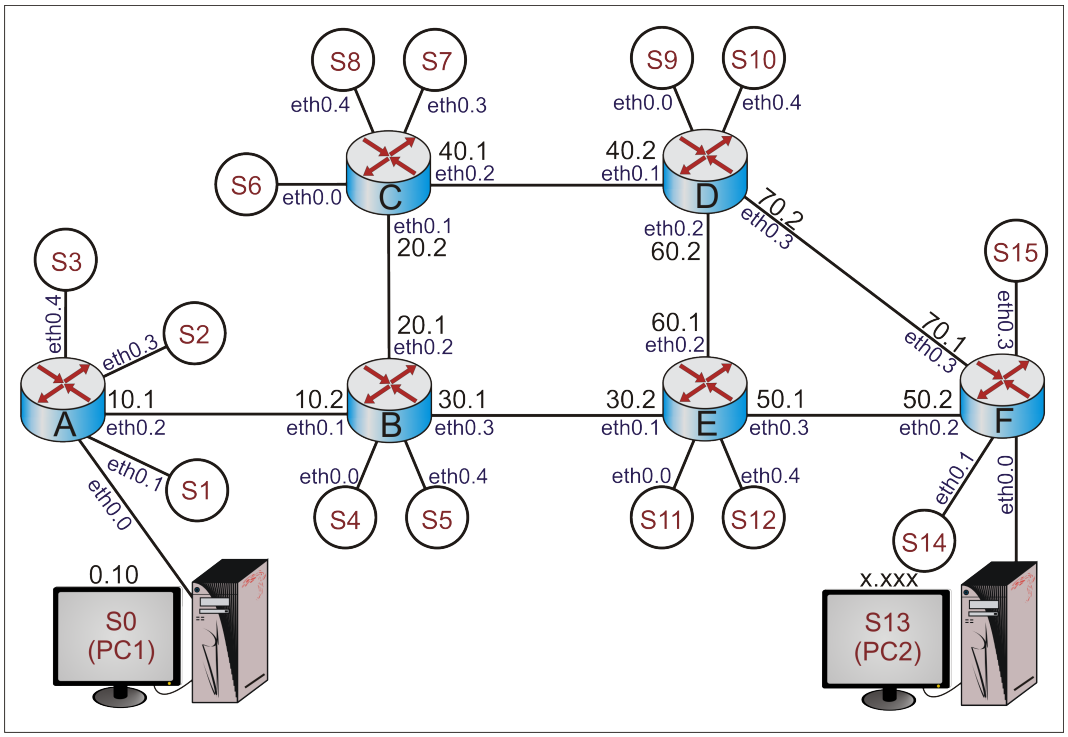
Druhá a třetí část bude zaměřena na dynamické směrování s použitím RIP a dále OSPF protokolu. Studenti budou seznámeni s implementací daného protokolu a po jejich zavedení budou testovat konektivitu a trasovat cestu přeposílaných paketů. Postupně budou zjišťovat zásadní rozdíly mezi těmito protokoly, například: jak ovlivní přenos paketů, od jednoho počítače k druhému, přerušení linky. Zařízení budou přístupné mezi počítači, takže odpojování bude prováděno manuálně. Tyto části rovněž seznámí studenty se stanovením metrik obou protokolů, a jaký vliv má na jejich výpočet například přenosová šířka pásma.

Mezi částmi úlohy bude důležité vždy restartovat, či nahrát původní konfiguraci daného zařízení (více v kapitole 3.1.3.

Na konci úlohy budou mít studenti vymezen čas pro kontrolní otázky k úloze.

2.4 Návrh topologie

Pro odlišení jednotlivých směrovacích metod bude potřeba jedné, či více smyček v síťové topologii, aby bylo možné k cíli putovat různými směry a v případě simulovaného výpadku linky, mohla být vybrána další redundantní cesta. Mezilehlé linky směrovačů budou mít rovněž různou přenosovou šířku pásma, což je prioritou pro výpočet metriky protokolu OSPF.



Obr. 2.3: Návrh topologie úlohy

V první části úkolu bude použita zjednodušená topologie v ohledu na složitost statické konfigurace. U druhého a třetího úkolu bude použita stejná topologie, ale již se všemi směrovači, aby studenti mohli efektivně porovnávat chování a změny mezi použitými směrovacími metodami. Topologie pro laboratorní úlohu, která splňuje předešlé požadavky, může nabírat vzhledu jako na obrázku 2.3. V tomto případě je mezi PC1 a PC2 k dispozici 4 různých cest. Popisky S0 – S15 značí koncové sítě.

Výstupem laboratorní úlohy je pochopení procesů a mechanizmů získání směrovacích informací výše uvedenými třemi metodami. Pro studenty budou přichystané konfigurační soubory, aby se nemuseli opakovaně zabývat stejným nastavením dalších směrovačů a přímo se věnovali podstatou úlohy, tedy směrováním. U protokolu OSPF bude nastavena různá šířka pásma mezi uzly tak, aby mezi počítači existovaly cesty s rozdílnou metrikou. Studenti se zaměří na faktory výběru cesty nejen při protokolu RIP a OSPF dle metriky, či administrativní vzdálenosti, ale i dle délky prefixu, kdy se bude jednat o konkrétnost sítě. Během postupu bude dále simulován výpadek linky manuálním odpojením, či externě za pomocí příkazu shutdown. Během toho bude studenty sledováno chování sítě (za použití příkazů *ping*, *traceroute*, atd. . .), a jak vzniklý problém vyřeší použitý směrový algoritmus.

3 REALIZACE LABORATORNÍ ÚLOHY

3.1 Sestavení pracoviště – počítače

Pro laboratorní úlohu jsou k dispozici 2 počítače s následujícími parametry, které jsou důležité k budoucímu nastavení virtuálního systému:

- **OS** Windows 7 Professional 64-bit (Build 7601),
- **procesor** Intel(R) Core(TM) i5-2400S CPU @ 2.50 GHz (4 CPUs),
- **paměť RAM** 8192 MB,
- **grafická karta** Intel(R) HD Graphics s vnitřní pamětí 1696 MB (podporované rozlišení 1280 × 1024 pro monitor HP LP1965 LCD Monitor),
- **HDD** o velikosti 731 GB (virtuálně rozdělený na disk C – 331 GB a na disk D – 400 GB)
- 2 **síťové karty**.

3.1.1 Virtuální systém

Veškeré předchozí testy byly prováděny na osobním notebooku s OS Windows 7 Professional, pro správnou kompatibilitu metodik řešení (hlavně skripty .vbs a PRTG síťový monitoring, který již nemá podporu pro Windows XP) bude tento systém použit i pro virtuální stroje.

Pro virtuální systém je na základě vlastností PC přidělena velikost paměti RAM 4096 MB, dále vytvořen dynamicky alokovaný virtuální pevný disk (.vdi) o velikosti 30 GB a v nastavení daného virtuálního systému jsou přiřazeny 2 jádra CPU. Pro lepší spolupráci s hardwarem PC je do virtuálního systému nahrán balíček „VBoxWindowsADDitions“ obsahující ovladače zařízení a systémové aplikace, které optimalizují operační systém pro lepší výkon a použitelnost. Pro propojení virtuálního systému se sítí pracoviště je nastaveno připojení síťové karty jako síťový most. Internet ve virtuálním systému není potřeba.

Na obou počítačích je nainstalován program Putty verze 0.63.0.0. U této verze je u připojení SSH vyžadována dodatečná interakce uživatele pouze při prvním připojení k cizímu zařízení. Počítač si následně uloží jeho identifikaci do souboru „známých hostů“ a při opětovném připojení počítač považuje zařízení jako důvěryhodné. Vstup je tedy povolen bez potřeby dalšího jednání uživatele. (novější verze Putty si identifikace zařízení neukládají, což by v budoucnu zabraňovalo správnému fungování skriptů, viz kapitola 3.1.3).

PC1 s virtuálním systémem BARS_Lx-PC1 bude sloužit k odchytávání paketů pomocí programu Wireshark (verze 2.2.5 64-bit) spolu s podprogramem WinPcap

(verze 4.1.3). Dále se z tohoto počítače budou spouštět skripty pro překonfiguraci všech směrovačů (viz dále v kapitole 3.1.3).

PC2 s virtuálním systémem BARS_Lx-PC2 je zřízen jako PRTG server a bude sloužit k monitorování celkových statistik směrovačů.

Po veškerém nastavení jsou virtuální disky systémů ve VirtualBoxu převedeny z normálního módu na neměnný ve Správci virtuálních médií. Po této změně je aktuální virtuální systém bez disku a je tedy potřeba náš používaný virtuální disk opět nahrát k systému v kartě Uložiště u řadiče SATA. Ve výsledku bude jakákoliv změna v systému, způsobená studenty, po restartu zpět v původním nastavení.

3.1.2 Zavedení PRTG do virtuálního systému PC2

Na stránkách `paessler.com` je volně ke stažení PRTG balíček. Po jeho rozbalení a následné instalaci ve virtuálním systému PC2 se na ploše zobrazí 2 nové spouštěče. PRTG Enterprise Console spustí PRTG v konzoli, druhý spouštěč PRTG Network Monitor otevírá PRTG ve webovém rozhraní, které je graficky líp zpracované. Pro budoucí účely bude tedy využíván pouze druhý spouštěč. Při prvotním spuštění se automaticky spustí průvodce nastavením. V nastavení *zařízení* je především zvolen interval skenování na nejméně možný a to 30 sekund. V nastavení účtu administrátora jsou změněny přihlašovací údaje na **Admin/PrtGmnH01**. Dále je v PRTG přidán studentský účet s omezenými právy jen pro čtení s přihlašovacími údaji **Student/Student2**. V poslední řadě je ve správě systému *uživatelského rozhraní* přenastaveno snímání živého grafu ze 120 hodnot na 240. Díky tomu budou vždy snímány poslední 2 hodiny, což pokryje čas pro celou laboratorní úlohu.

Správa zařízení a senzorů

Strom všech zařízení a senzorů je zobrazen po spuštění „Zařízení“ na vrchní liště. Strom začíná od Sondy Local Probe, dále jsou přidávány zařízení dle jejich IP adresy, následně se na zařízení přidávají senzory. Typy senzorů, které budou v úloze využívány jsou Ping (testuje dostupnost zařízení), SNMP provoz (měří vstupní i výstupní přenosovou rychlost) a Zatížení SNMP CPU.

3.1.3 Skripty v jazyce VBScript

Před každou konfigurací metody směrování bude potřeba přivést každý směrovač do původního stavu, jelikož například při sestavení statické cesty k druhému počítači bude tato cesta vždy upřednostněna před směrováním pomocí RIP či OSPF protokolu. Navíc, nutnost opakované konfigurace všech směrovačů rovněž není úplně optimální a úplně postačí, aby každý student měl na starost 2 směrovače z „jeho

strany“ připojení. Kdyby tyto úkony měli studenti provádět sami, vypracovat úlohu do 90 minut by bylo nemožné, navíc by zde byla velká šance způsobit nějakou chybu, která by ovlivnila průběh úlohy (například přepis špatné konfigurace, zapomenout restartovat nějaký směrovač, atd. . .). Jakýkoli chybný přepis konfigurace je pak špatně dohledatelný, jelikož je potřeba kontrola 3 konfiguračních souborů postupně na každém směrovači v linuxovém prostředí.

Celkem bude potřeba tří skriptů. První bude všechny směrovače nastavovat do původního nastavení (přepis konfigurací OSPF). Druhý bude mazat konfigurace statických záznamů a nahrávat konfigurace ohledně RIP protokolu u směrovačů C a D. Třetí skript bude opět mazat konfigurace a záznamy předchozího úkolu ohledně RIP směrování a nahraje konfigurace potřebné k fungování OSPF na směrovače C a D. Studenti tak budou konfigurovat vždy A, B a E, F směrovače.

Ve Windows 7 se dle těchto preferencí nabízí jako snad jediné řešení jazyk VBScript. Pro svoji funkčnost potřebuje pouze zápis v textovém dokumentu, který je následně otvírán v programu Microsoft Windows Based Script Host. V každém VBScriptu bude použito úvodní přihlášení ke směrovači RouterA jako ve výpisu kódu 3.1.

Výpis 3.1: Přihlášení na směrovač RouterA přes Putty

```
1  Option Explicit
2  On Error Resume Next
3  Dim RouterSHell
4  Dim PUTTYPROFILE
5  PUTTYPROFILE = "RouterA"
6  set RouterShell=CreateObject("WScript.Shell")
7  RouterShell.run "putty.exe -load " & chr(34) & "RouterA" & chr(34)
8  WScript.Sleep 1500
9  RouterShell.SendKeys ("root")
10 RouterShell.SendKeys ("{Enter}")
11 WScript.Sleep 300
12 RouterShell.SendKeys ("linksys")
13 RouterShell.SendKeys ("{Enter}")
14 WScript.Sleep 500
```

Řádek 1 deklaruje všechny proměnné VBScriptu. Příkaz *Dim* deklaruje úložný prostor pro proměnné, na řádku 5 je definován profil, který je předem v programu Putty uložen. Tento profil obsahuje přihlášení na rozhraní 10.10.0.1 s připojením přes SSH. Řadek 7 spouští program Putty a v něm načítá dříve zmíněný profil, *chr(34)* představuje uvozovky dle 34. symbolu ASCII. Příkaz *WScript.Sleep x* definuje pozastavení skriptu v milisekundách. Tento příkaz je velmi podstatný pro správné fungování kódu. Například při přihlášení je potřeba čekat na součinnost směrovače, kdy je potřeba zadávat přihlašovací údaje (**root**/**linksys**) až po nabídnutí jejich

vložení ve tvaru *Login:* a *Password:*. Rovněž při příliš rychlém restartování všech směrovačů dojde k jejich zaseknutí, odblokování je pak možné pouze odpojením od zdroje napájení a po vymazání všech mezipamětí je znovu zapojit. Jednotlivé řetězce kódu jsou na směrovač posílány příkazem *SendKeys*.

Výpis 3.2: Přihlášení ze směrovače RouterA na RouterB

```
15 RouterShell.SendKeys ("ssh 10.10.10.2")
16 RouterShell.SendKeys ("{Enter}")
17 WScript.Sleep 2000
18 RouterShell.SendKeys ("linksys")
19 RouterShell.SendKeys ("{Enter}")
20 WScript.Sleep 500
```

Výpis 3.2 posílá příkazy na RouterA a pomocí ssh se dále připojuje na RouterB. Již jsme přihlášení jako uživatel **root**, proto je po nás vyžadováno jen vložení hesla. Dle těchto šesti řádků se dále postupně přihlašujeme na další směrovače C, D, E a F.

Výpis 3.3: Přepis předešlé konfigurace

```
47 RouterShell.SendKeys ("cp \etc\quagga\zaloha\F01_ripd.conf \etc\
       quagga\ripd.conf")
48 RouterShell.SendKeys ("{Enter}")
49 WScript.Sleep 500
```

Tato část výpisu 3.3 má za úkol připravit konfigurace směrovačů pro nadcházející úkol. Pokud by studenti pracovali přesně podle návodu, postačilo by vždy jen restartování procesu Quagga. Do budoucna je ale předvídáno, že studenti mohou v terminálu lehce změnit obsah konfiguračního souboru například příkazem *copy running-config startup-config*. Toto nastavení by se následně muselo ručně přepsat zpět v prostředí OpenWRT, jelikož terminál neumožňuje mazat *startup* konfiguraci. Proto jsou při každém spuštění skriptu rovnou přepisovány všechny konfigurace modulů, se kterými je v předešlém úkole manipulováno.

Pro tento krok je na všech směrovačích vytvořen adresář „zaloha“ s cestou `\etc\quagga\zaloha`, kde jsou uloženy konfigurační zálohy pro jednotlivé směrovací moduly. Tyto zálohy jsou ve tvaru *XYY_ZZZZ.conf*, kde *X* značí název směrovače (A – F), *YY* značí předkonfigurovanou zálohu ve tvaru 01 a nebo zálohu s již nakonfigurovaným směrováním ve tvaru 02, každý směrovací modul má svůj vlastní konfigurační soubor, *ZZZZ* označuje, pro který směrovací modul je záloha určena. Na řádku 45 je použit pro přepis konfigurace soubor *F01_ripd.conf*. Označení znamená, že záloha je určena k předkonfigurování směrovače F u protokolu RIP. Využití

této zálohy je zřejmé pro další úkol konfigurace OSPF, jelikož všechny směrovače, kde byl nakonfigurován RIP protokol, je potřeba nastavit do původního stavu.

Výpis 3.4: Restart procesu Quagga

```
50 RouterShell.SendKeys ("\etc\init.d\quagga restart")
51 RouterShell.SendKeys ("{Enter}")
52 WScript.Sleep 7000
53 RouterShell.SendKeys ("exit")
54 RouterShell.SendKeys ("{Enter}")
55 WScript.Sleep 100
```

Aby se projevily nastavení konfiguračního souboru, je potřeba restartovat zařízení nebo restartovat běžící proces Quagga. Jelikož tyto směrovače nejsou určené ke každodennímu opakovanému restartování, bude použito druhé možnosti. Restartování zprostředkovává 48. řádek ve výpisu 3.4. Bohužel je potřeba počkat na dokončení restartování procesu 7 sekund, poté zadáním příkazu *exit* se dostaneme na předešlý směrovač (E). Kombinace výpisu 3.3 a výpisu 3.4 se opakuje až do opuštění směrovače A (jen na směrovačích D a C budeme navíc přepisovat pomocí záloh D02 a C02, což představuje nakonfigurování směrového mechanizmu nadcházejícího úkolu).

Celý proces trvá přibližně 60 s, během této doby nesmí být používána myš či klávesnice, protože příkaz *SendKeys* posílá příkazy jen do právě otevřeného okna. V případě překliknutí do jiného okna příkazy nebudou posílány do programu Putty.

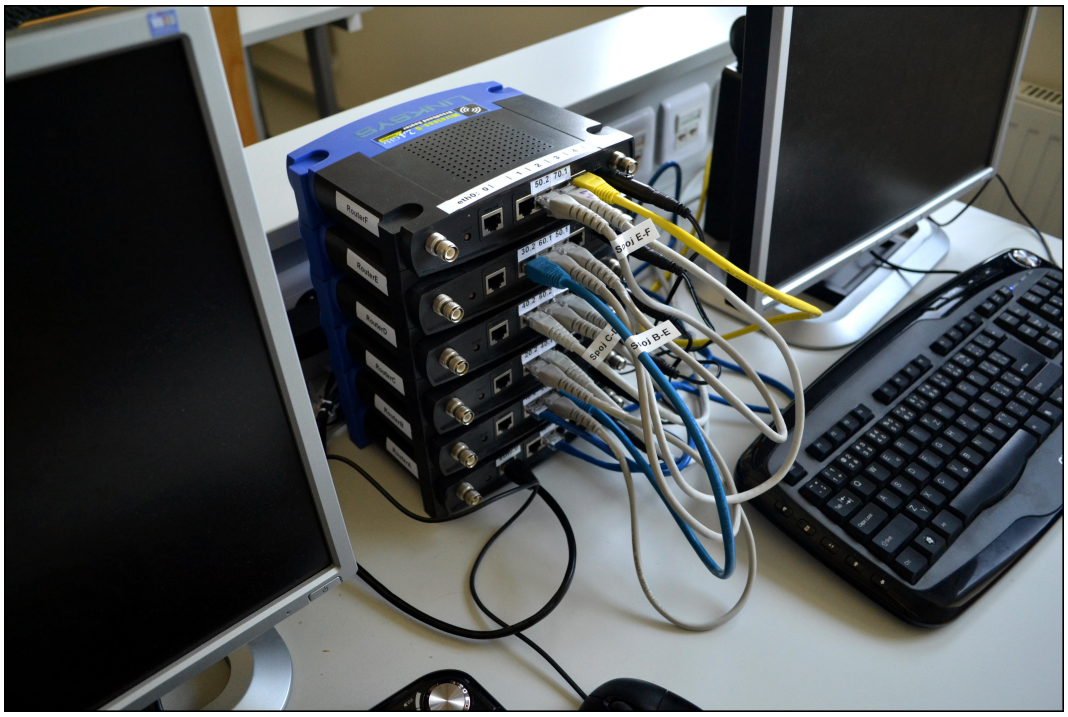
3.2 Sestavení pracoviště – směrovače

Všechny směrovače jsou umístěny mezi počítači PC1 a PC2, jsou studentům lehce přístupné a mohou tak jednoduše provádět přepojování svých počítačů a nebo simulovat výpadek linky jejím odpojením. Kabeláž, se kterou je často manipulováno je barevně odlišena či popsána. Fyzické sestavení laboratoře je znázorněno na obrázku 3.1.

Další podkapitoly se zabývají softwarovou částí směrovačů a problematikou spojenou s jejím nastavením.

3.2.1 Nastavení portů a IP adres

Směrovač Linksys WRT54GL má 1 port WAN určený ke komunikaci s ostatními směrovači a ostatní 4 porty se chovají jako obyčejný přepínač. Abychom v laboratorní úloze mohli směrovače navzájem propojovat budeme potřebovat více než 1 port k tomu určený. Proto jsou porty vnitřně rozděleny na VLANy, ty jsou pak



Obr. 3.1: Uspořádání pracoviště

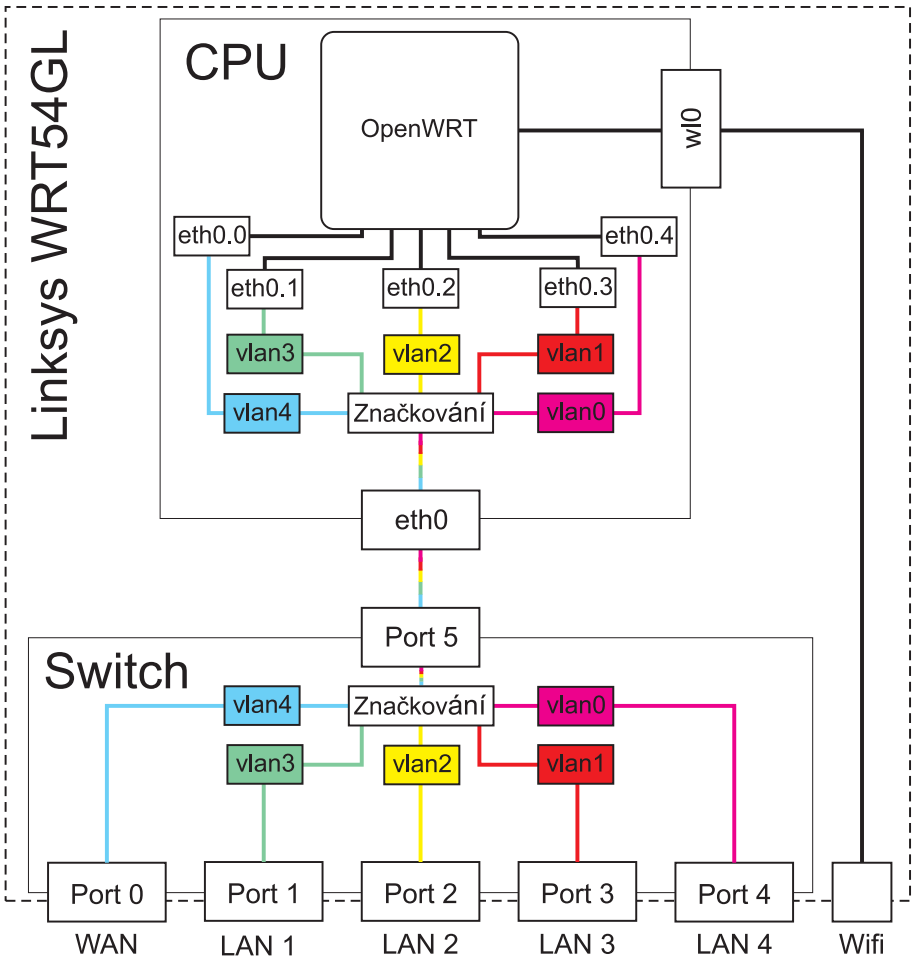
značkovány a poslány do procesoru, kde s příchozími informacemi již pracuje systém OpenWRT na základě nastavení. Tato konfigurace je provedena v souboru `\etc\config\network`.

Výpis 3.5: Rozdělení portů na VLANy

```
7  config 'switch_vlan'
8          option 'device' 'eth0'
9          option 'vlan' '0'
10         option 'ports' '4 5t'
11
12 config 'switch_vlan'
13         option 'device' 'eth0'
14         option 'vlan' '1'
15         option 'ports' '3 5t'
16
17 config 'switch_vlan'
18         option 'device' 'eth0'
19         option 'vlan' '2'
20         option 'ports' '2 5t'
21
22 config 'switch_vlan'
23         option 'device' 'eth0'
24         option 'vlan' '3'
```

```
25         option 'ports' '1 5t'
26
27 config 'switch_vlan'
28         option 'device' 'eth0'
29         option 'vlan' '4'
30         option 'ports' '0 5t'
```

Rozdělení portů na VLANy definuje výpis 3.5 příkazem *switch_vlan*, dle tohoto nastavení si lze vnitřní strukturu směrovače představit jako tomu je na obrázku 3.2 [9].



Obr. 3.2: Vnitřní architektura směrovače Linksys WRT54GL

Od řádku 32 následuje nastavení IP adres jednotlivých rozhraní, výpis 3.6 znázorňuje nastavení rozhraní eth0.0 směrovače RouterA. IP adresy lze nastavovat i přes modul Zebra, avšak to by umožnilo studentům je pozměnit a poškodit tak funkčnost úlohy. Síť laboratorní úlohy má k dispozici adresový rozsah 10.10.0.0/16. Podsítě

Výpis 3.6: Definice rozhraní eth0.0 směrovače RouterA

```
32 config 'interface' 'eth0'
33         option 'proto' 'static'
34         option 'ifname' 'eth0.0'
35         option 'ipaddr' '10.10.0.1'
36         option 'netmask' '255.255.255.128'
```

mezi směrovači mají pevnou délku prefixu 30, zato koncové sítě pracují s proměnlivou délkou masky a jsou v rozsahu 10.10.0.0/21.

3.2.2 DHCP

Na všech volných rozhraních směrovačů je nastaven DHCP server (na obrázku 2.3 jsou sítě s volným rozhraním označeny jako S0 – S15). Studenti tak budou moci v průběhu úlohy různě přepojovat své počítače a testovat mezi sebou konektivitu z různých sítí a tím ověřit správnost konfigurace směrovačů. Nastavení DHCP se provádí v souboru `\etc\config\dhcp`.

Výpis 3.7: Konfigurace DHCP na směrovači RouterA

```
1  config 'dnsmasq'
2          option 'boguspriv' '1'
3          option 'localise_queries' '1'
4          option 'expandhosts' '1'
5          option 'readethers' '1'
6          option 'leasefile' '/tmp/dhcp.leases'
7          option 'resolvfile' '/tmp/resolv.conf.auto'
8          option 'rebind_protection' '0'
9
10 config 'dhcp'
11         option 'interface' 'eth0'
12         option 'start' '2'
13         option 'limit' '125'
14         option 'leasetime' '1h'
15
16 config 'dhcp'
17         option 'interface' 'eth1'
18         option 'start' '130'
19         option 'limit' '61'
20         option 'leasetime' '1h'
21
22 config 'dhcp'
23         option 'interface' 'eth3'
24         option 'start' '194'
25         option 'limit' '29'
```

```
26          option 'leasetime' '1h'
27
28  config 'dhcp'
29          option 'interface' 'eth4'
30          option 'start' '226'
31          option 'limit' '29'
32          option 'leasetime' '1h'
33
34  config host
35          option ip '10.10.0.10'
36          option mac '08:00:27:33:ab:a8'
37          option name 'PC1'
```

Ukázka konfigurace DHCP je ve výpisu 3.7. Příkaz *dnsmasq* specifikuje soubor pro zápis vypůjčených IP adres v položce *leasefile*, ostatní položky představují defaultní nastavení pro DNS. Následuje nastavování jednotlivých DHCP serverů na volná rozhraní příkazem *dhcp*. Položka *interface* určuje rozhraní DHCP serveru, *start* je hodnota posledního oktetu IP adresy, *limit* je počet možných vypůjček snížen o hodnotu 1, *leasetime* je doba výpůjčky. Příkaz *host* specifikuje přidělení statické adresy počítači s danou MAC adresou.

3.2.3 Problematika spojená s propojením koncových uživatelů

Po nakonfigurování směrovací politiky v experimentální síti nelze ověřit spojení za pomocí ICMP protokolu mezi koncovými uživateli. Tyto příchozí zprávy blokují 2 firewally. Na síťové kartě, která má na starost připojení do experimentální sítě, je povolena pouze linková vrstva, aby zprávy neprocházely firewallem samotného počítače. Díky tomu se příchozí zprávy rovnou předají virtuálnímu systému. Ve virtuálním systému je upraveno pravidlo firewallu pro ICMPv4, aby povolovalo libovolné vzdálené i místní IP adresy.

3.2.4 Problematika spojená s modulem RIP

Instalovaný balíček *quagga-ripd – 0.98.6-5* obsahuje spoustu chyb. Primárním problémem bylo zahazování aktualizačních RIP paketů. Po vyzkoušení všech dostupných návodů na Internetu byla závada dále konzultována se zaměstnanci společnosti Red Hat, kteří se zabývají vývojem softwaru Quagga. Uvedené verzi skončila podpora před 6 lety, proto jako jediné řešení doporučili přeinstalovat verzi na vyšší. Momentální verze OpenWRT 10.03 může být přeinstalována pouze na verzi 10.03.1. Vyšší verze nemohou být použity, jelikož směrovač WRT54GL má paměť RAM pouze o velikosti 16 MB.

Není zajištěno, aby vylepšení verze systému na 10.03.1 vyřešilo zmíněnou závadu, proto bylo pátráno hlouběji. Logování v případě RIP protokolu nebylo funkční, jelikož skupina Quaqqa neměla právo přistupovat a zapisovat do logovacího souboru. Příkazem *chmod* byly práva změněny. Výpis 3.8 znázorňuje zaznamenanou aktivitu modulu RIP.

Výpis 3.8: Aktivita modulu RIP

```
RECV packet from 10.10.50.2 port 520 on eth0.3
RECV RESPONSE version 2 packet size 44
   10.10.70.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
   192.168.2.0/24 -> 0.0.0.0 family 2 tag 0 metric 1
RIPv2 dropped because authentication enabled
```

Příčina zahazování je tedy způsobena autentizací, která je defaultně v modulu RIP zapnuta. Bez autentizace jsou RIP pakety zahazovány. Vypnutí tohoto autentizačního módu není možné, jedná se o bug 262 – arh200604-2. Jeho oprava je provedena ve verzi balíčku 0.99.x, který je dostupný v OpenWRT 10.03.1. Příjem RIP paketů byl nakonec zprovozněn nastavením autentizace na všech směrovačích se stejným klíčovým řetězcem. Ve výpisu 3.9 je znázorněno řešení pro směrovač RouterA.

Výpis 3.9: Přednastavení modulu RIP

```
hostname RouterA
password zebra

key chain ka1
 key 1
  key-string 234

interface eth0.2
 ip rip authentication key-chain ka1

line vty
```

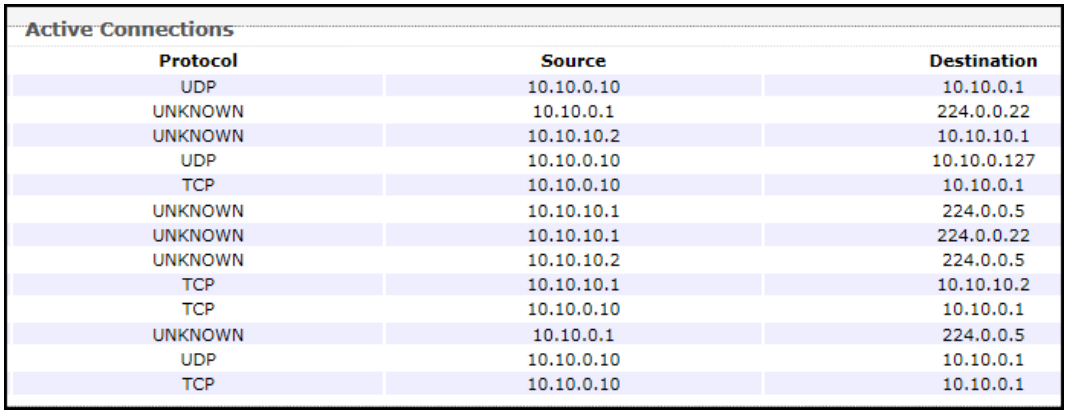
Možnost konfigurace RIP protokolu neumožňuje možnosti sumarizovat směrové záznamy, z tohoto důvodu nebude v úloze sumarizování adres zavedeno do postupu řešení.

3.2.5 Problematika spojená s modulem OSPF

Všechny směrovače jsou mezi sebou přímo připojeny, typ sítě mezi nimi by tedy měl být Point to Point, ale jelikož je Linksys WRT54GL fyzicky z velké části stále switch, OSPF detekuje tyto spojení jako Broadcast Multi Access (ve skutečnosti by ale měly být typu Point to Point). V důsledku toho se volí směrovače DR a BDR a tím

pádem je vyžadovaná i specifikace *router-id*. Problém pokračuje tím, že všechny směrovače mají nastavenou stejnou adresu loopback (využívaná pro připojení se na moduly deamonů), takže *router-id* se musí nastavovat ručně. Bez stanovení *router-id* si nebudou sousedé rozumět a směrování nebude funkční.

Další odlišností je konfigurace tohoto protokolu, kdy se sítě nespecifikují podle inverzní masky, ale podle velikosti prefixu (avšak výhodou této odlišnosti je jednodušší konfigurace). Modul OSPF, stejně jako modul RIP, neumožňuje sumarizaci adres.



Active Connections

Protocol	Source	Destination
UDP	10.10.0.10	10.10.0.1
UNKNOWN	10.10.0.1	224.0.0.22
UNKNOWN	10.10.10.2	10.10.10.1
UDP	10.10.0.10	10.10.0.127
TCP	10.10.0.10	10.10.0.1
UNKNOWN	10.10.10.1	224.0.0.5
UNKNOWN	10.10.10.1	224.0.0.22
UNKNOWN	10.10.10.2	224.0.0.5
TCP	10.10.10.1	10.10.10.2
TCP	10.10.0.10	10.10.0.1
UNKNOWN	10.10.0.1	224.0.0.5
UDP	10.10.0.10	10.10.0.1
TCP	10.10.0.10	10.10.0.1

Obr. 3.3: Aktivní spojení na směrovači RouterA

Aby si studenti mohli prohlédnout obsahy nejen paketů Hello, ale například i LSU, LSP, DBD, je potřeba, aby se virtuální systém choval jako směrovač s nakonfigurovaným OSPF protokolem. Pro tuto potřebu je ale vyžadován systém Windows Server 2008 a vyšší. Další možností je nastavit firewall tak, aby přesměrovával provoz OSPF na počítač. Po různých nastaveních toto řešení nebylo úspěšné, jelikož příkaz *redirect* vyžaduje specifikaci protokolu, avšak samotný protokol OSPF vidí systém OpenWRT jako neznámý (Unknown), viz obrázek aktivních spojení 3.3. Jediným řešením by mohlo být propojení směrovačů a počítače rozbočovačem, komunikace mezi směrovači by tak byla posílána i na počítač. Rozbočovač k úloze ale není k dispozici.

3.3 Testování laboratorní úlohy

Po zhotoveném návodu k laboratorní úloze byla testována jeho časová náročnost a porozumění částem návodu. K testování byli vybráni studenti s minimální znalostí problematické tématiky směrování v datových sítích. S přípravou 45 minut byli studenti schopni zvládnout úlohu přes ztížené podmínky a chyby v návodu za 95 minut. Na základě jejich postupu byl návod upraven a více přizpůsoben k lepšímu

pochopení a efektivnímu řešení dílčích kroků. Chyby, se kterými se studenti během postupu řešení setkali, byly odstraněny.

Sledovaná data u síťového monitoringu PRTG nebyly jednoznačná. Cílem zavedení tohoto nástroje bylo ukázat vyšší využití CPU u OSPF než u protokolu RIP a naopak vyšší zatížení sítě provozními pakety RIP než u OSPF. S ohledem na velikost laboratorní sítě bylo zatížení CPU směrovačů nerozeznatelně nízké (většinou 1%). Provoz v této síti byl ovlivněn pakety SNMP. Omezení senzorů jen na jeden port tento problém značně vyřešil, avšak data byla stále špatně čitelná. Rovněž provoz velmi ovlivňuje psaní do terminálů směrovačů, stejně tak ICMP zprávy příkazů *ping* a *tracert*. Ve finále bylo pro studenty těžké určit klidový provoz směrového protokolu. V závislosti na časovou náročnost úlohy a výše zmíněné potíže, je nástroj PRTG z úlohy vyloučen.

Po úpravě návodu k úloze a odstranění nástroje PRTG by studenti 2. ročníku teleinformatiky měli být schopni zvládnout úlohu do 90 minut.

4 ZÁVĚR

Bakalářská práce se zabývá rozborem problematiky směrování v datových sítích. V průběhu práce je vysvětlována komunikace na spojové a síťové vrstvě, jakým způsobem je určována cesta napříč síťovými uzly a co všechno je potřeba k doručení dat cíli.

Směrování nabízí 2 metody zápisu směrovacích tabulek a to statické nebo dynamické. Obě metody mají své výhody i nevýhody popsány v kapitole 1.4 spolu s algoritmy dynamického směrování, které představují rodiny pro dynamické směrové protokoly, z nichž vyplývají jejich vlastnosti.

Na základě popsaných a prozkoumaných možností směrování byla navržena laboratorní úloha s dílčími úkoly včetně topologie a struktury laboratorního pracoviště. Obsahem úlohy je zajistit komunikaci v síti pomocí statického a následně i dynamického směrování. Do úlohy byly zahrnuty dynamické protokoly RIP a OSPF pro svá velmi odlišná chování, jelikož jsou oba z jiné skupiny protokolů a tedy fungují na úplně odlišných principech. Postupem řešení budou studenti, u těchto jednotlivých směrovacích technik, analyzovat vlastnosti a rozdílné chování v síti.

Z počátku realizace bylo potřebné se především naučit pracovat v operačním systému Linux, ze kterého vychází i ovládání systému OpenWRT. Následně, až po tomto učinění bylo možné přistoupit ke konfiguracím různých částí směrovače. Následná realizace pracoviště, a problematika s ní spojená, je popsána v kapitole 3. Je zde kladen důraz na přípravu pracoviště z pohledu počítačů a následně i směrovačů. V průběhu nastavení směrovačů se autor práce setkal s různou problematikou, u které bylo nutné prozkoumat veškeré možnosti a zvolit optimální řešení, pokud to daná situace umožňovala.

V rámci této práce byly vytvořeny skripty, které na začátku každého úkolu v laboratorní úloze nastavují směrovače do původního stavu a zároveň ulehčují studentům práci nakonfigurováním dvou směrovačů. Tím je taktéž získán čas, podstatný pro dokončení úlohy ve stanoveném čase 90 minut. K tomuto přenastavení jsou vytvořeny konfigurační zálohy, umístěné v odlišném adresáři než původní konfigurace, se kterými pracuje proces Quagga.

V poslední části této práce byl vypracován návod k laboratorní úloze, který byl upraven tak, aby byl časově zvládnutelný ve výukových hodinách laboratoře předmětu Architektura sítí.

LITERATURA

[1] BALCHUNAS, A. *Cisco CCNP Routing Study Guide.* In: Router Alley [online]. Michigan, 2012 [cit. 2016-12-06]. Dostupné z URL: <http://www.routeralley.com/completed/ccnp_routing_studyguide.pdf>.

[2] BLACK, U. D. *IP routing protocols: RIP, OSPF, BGP, PNNI and Cisco routing protocols.* Upper Saddle River, NJ: Prentice Hall, 2000, 287 p. ISBN 01-301-4248-4.

[3] BOUŠKA, P. *Cisco – Router Switching metody a související termíny – CAM, FIB, CEF.* Samuraj [online]. 2009 [cit. 2016-11-02]. Dostupné z URL: <http://www.samuraj-cz.com/clanek/cisco-router-switching-metody-a-souvisejici-terminy-cam-fib-cef>.

[4] BOUŠKA, P. *Cisco Routing 1 – obecné vlastnosti směrovacích protokolů.* Samuraj [online]. 2009 [cit. 2016-11-02]. Dostupné z URL: <http://www.samuraj-cz.com/clanek/cisco-routing-1-obecne-vlastnosti-smerovacich-protokolu>.

[5] BOUŠKA, P. *TCP/IP – Routing – směrování.* Samuraj [online]. 2007 [cit. 11. 2. 2016]. Dostupné z URL: <http://www.samuraj-cz.com/clanek/tcpip-routing-smerovani>.

[6] BOUŠKA, P. *Víte, jak pracuje router?.* Samuraj [online]. 2010 [cit. 2016-11-02]. Dostupné z URL: <http://www.samuraj-cz.com/clanek/vite-jak-pracuje-router>.

[7] CARRERA, B. *QUAGGA – The Easy Tutorial – How to use Quagga.* Openmaniak [online]. 2010 [cit. 2017-03-25]. Dostupné z URL: <https://openmaniak.com/quagga_tutorial.php>.

[8] Cisco Engineers. *Catalyst 6500/6000 Switches ARP or CAM Table Issues Troubleshooting.* Cisco [online]. Říjen 27, 2009 [cit. 2017-03-09]. Dostupné z URL: <http://www.cisco.com/c/en/us/support/docs/ip/address-resolution-protocol-arp/117398-qanda-arp-timeout-00.html>.

[9] FOXSELL. *Linksys WRT54G, WRT54GL, WRT54GS.* OpenWRT Wiki [online]. 2016 [cit. 2017-05-25]. Dostupné z URL: <https://wiki.openwrt.org/toh/linksys/wrt54g>.

[10] GRYGÁREK, P. *Směrovací protokol BGP.* Počítačové sítě [online]. Ostrava, 2016 [cit. 2016-12-06]. Dostupné z URL: <http://www.cs.vsb.cz/grygarek/SPS/lect/BGP/BGP.html>.

[11] GRYGÁREK, P. *Směrovací protokol OSPF.* Počítačové sítě [online]. Ostrava, 2016 [cit. 2017-03-25]. Dostupné z URL: <http://www.cs.vsb.cz/grygarek/SPS/lect/OSPF/ospf.html>.

[12] MÁCHA, T. *Dynamic Metric in OSPF Networks.* Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, 2015. 119 p. Supervised by doc. Ing. Vít Novotný, Ph.D.

[13] MEDHI, D. and RAMASAMY, K. *Network Routing: Algorithms, Protocols, and Architectures.* San Francisco: Morgan Kaufmann Publishers, c2007, 768 p. Morgan Kaufmann Series in Networking. ISBN 978-0-12-088588-6.

[14] Novotný, V. *Architektura sítí.* Brno, ČR: VUT v Brně, 2012, 152 s. ISBN 80-864-9755-0.

[15] ROZSYPAL, O. *Analýza směrovacích protokolů.* Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 77 s. Vedoucí bakalářské práce Ing. Anna Kubánková, Ph.D.

[16] SEIFERT, R. *The Switch Book: The Complete Guide to LAN Switching Technology.* New York: John Wiley, 2000, 698 p. ISBN 0-471-34586-5.

[17] SHINDER, D. L. *Počítačové sítě: Nepostradatelná příručka k pochopení síťové teorie, implementace a vnitřních funkcí.* Praha: SoftPress, c2003, 752 s. Cisco systems. ISBN 80-864-9755-0.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ABR	Area Border Router
AD	Administrative Distance
ARP	Address Resolution Protocol
AS	Autonomous System
ASBR	Autonomous System Border Router
ASCII	American Standard Code for Information Interchange
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CAM	Content Addressable Memory
CPU	Central Proccessing Unit
DBD	Database Description
DBR	Backup Designated Router
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DR	Designated Router
DUAL	Diffusing update algorithm
DVA	Distance Vector Algorithm
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
GPS	Global Positioning System
ICMP	Internet Control Message Protocol
ID	IDentification
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IOS	Internetwork Operating System
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
LAN	Local Area Network
LSA	Link State Algorithm / Link State Advertisement
LSAck	Link State Acknowledgement
LSAs	Link State Advertisements
LSP	Link State Packet
LSR	Link State Request

LSU	Link State Update
MAC	Media Access Control
MD5	Message Diggest 5
MPLS	MultiProtocol Label Switching
MTU	Maximum transmission unit
NBMA	Non-broadcast Multi Access
NDP	Neighbour Discovery Protocol
NSSA	Not So Stubby Area
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPFv2	Open Shortest Path First version 2
OSPFv3	Open Shortest Path First version 3
PRTG	Paessler Router Traffic Grapher
PVA	Path Vector Algorithm
RAM	Random Access Memory
RIP	Routing Information Protocol
RIPv1	Routing Information Protocol version 1
RIPv2	Routing Information Protocol version 2
RIPng	Routing Information Protocol next generation
SNMP	Simple Network Management Protocol
SPF	Shortest Path First
SSH	Secure Shell
TTL	Time To Live
VLAN	Virtual Local Area Network
VLSM	Variable-Length Subnet Mask
WAN	Wide Area Network
WLAN	Wireless Local Area Network

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje elektronickou verzi bakalářské práce.

- `/` ..... kořenový adresář přiloženého CD
  - `Skripty` ..... skripty v jazyce VBScript
    - `01_Static.vbs`
    - `02_RIP.vbs`
    - `03_OSPF.vbs`
  - `Zalohy` ..... zálohy všech směrovačů
    - `Pocatecni konfigurace` ..... před-konfigurační výpisy směrovačů
      - `DHCP.txt`
      - `Network.txt`
      - `OSPFd01.txt`
      - `RIPd01.txt`
      - `ZEBRAd01.txt`
    - `Koncove konfigurace` ..... výpisy koncových konfigurací směrovačů
      - `OSPFd02.txt`
      - `RIPd02.txt`
      - `ZEBRAd02.txt`
  - `BP_Tomas Stodulka.pdf` ..... hlavní dokument práce
  - `Laboratorni protokol.pdf` ..... návod k laboratorní úloze

B NÁVOD K LABORATORNÍ ÚLOZE

Pro možnosti budoucích úprav je vyhotovený laboratorní protokol ve formátu *DOCX*, tento formát je však z důvodu studentských úprav odevzdán pouze vedoucímu předmětu.

1 SMĚROVÁNÍ V DATOVÝCH SÍTÍCH

Cíl

Cílem laboratorní úlohy je seznámit studenty s problematikou směrování v datových sítích u statických a dynamických směrových technik, konkrétně s protokoly RIP a OSPF. Absolvováním této laboratorní úlohy student získá základní znalosti o mechanizmech zajištění směrových informací a dokáže stručně popsat výhody a nevýhody implementované metody směrování.

Vybavení pracoviště

6× směrovač Cisco Linksys WRT54GL v1.1, 2× pracovní stanice s virtuálními systémy MS Windows 7 Proffesional, Wireshark, Putty.

Úkoly

1. Projděte si teoretický úvod a seznamte se s pracovištěm úlohy.
2. Zajistěte komunikaci v síti statickým směrováním.
3. Zajistěte komunikaci v síti směrovým protokolem RIP, simulujte výpadek linky a analyzujte jeho celkové chování.
4. Zajistěte komunikaci v síti směrovým protokolem OSPF, simulujte výpadek linky a analyzujte jeho celkové chování.
5. Na základě zjištěných skutečností odpovědět na kontrolní otázky.

1.1 Teoretický úvod

1.1.1 Směrování

Směrování zajišťuje komunikaci mezi různými sítěmi a využívá k tomu první 3 vrstvy ISO/OSI. Provoz je orientován na základě směrových tabulek směrovačů, které slouží k jistému zmapování síťové topologie. Směrová tabulka primárně obsahuje **IP adresy** dostupných sítí s **maskou**, **bránu** (IP adresa dalšího směrovače na cestě k cíli, též někdy označována jako Next-hop adresa) a **výstupní rozhraní** určující směr k dané dostupné síti. Dále daný řádek obsahuje bránu, metriku a způsob získání směrového záznamu, ten může být statický, dynamický, nebo automatický (přímo připojené sítě po připojení a nastavení rozhraní směrovače).

Ve směrové tabulce může být spousta možných cest k cíli, výběr té nejlepší určují tyto 3 aspekty:

- **Délka prefixu** – je první prioritou výběru cesty, kde upřednostňujeme ty s největší maskou.
- **Administrativní vzdálenost** – představuje důvěryhodnost daného směrového protokolu. V případě, že je směrovač konfigurován více směrovými protokoly, vybere se mezi nimi na základě nejnižší administrativní vzdálenosti viz následující tab. 1.

Tab. 1: Tabulka hodnot administrativních vzdáleností

Zdroj cesty	Administrativní vzdálenost
Přímo připojené rozhraní	0
Statická cesta	1
OSPF	110
RIPv1, RIPv2	120

- **Metrika** – pokud existuje více cest do stejné sítě zajištěných jedním směrovým protokolem, směrovač vybere tu nejlepší na základě nejmenší hodnoty metriky.

1.1.2 Statické směrování

Zápis do směrové tabulky je spravován ručně administrátorem, kde pro plnou konektivitu musí být nakonfigurovány cesty do všech vzdálených sítí na všech zařízeních. Taková konfigurace bývá náročná a ve větších sítích takřka nemožná, proto má statické směrování využití jen v menších a jednodušších sítích.

Staticky lze taky nakonfigurovat implicitní cestu, která se používá v případě, pokud neexistuje žádná jiná cesta k cíli. Její zápis je vždy ve tvaru 0.0.0.0/0 s kombinací Next-hop adresy (adresa směrovače, na který má být odesílaný provoz). Používá se zpravidla ke směrování do Internetu.

Sumarizace sítí

Sumarizace, neboli agregace, je proces zápisu více směrových informací do jedné, čímž zmenšíme počet směrových záznamů a tedy i zrychlíme výkon, jelikož při určování cesty se prochází celá směrová tabulka. (se sumarizací adres se stejně jako zde můžete setkat v **laboratorní úloze č. 1, úkol 2c**).

1.1.3 Dynamické směrování

Vzhledem k rozsáhlosti většiny dnešních datových sítí je statická konfigurace a údržba směrovačů administrátorem nemožná. Z tohoto důvodu se zavedlo dynamické směrování, kde je zásah administrátora vyžadován nejvíce při prvotní konfiguraci protokolu směrovače, kde administrátor přidává sousední sítě, s kterými má směrovač komunikovat.

Pomocí směrových protokolů spolu směrovače vzájemně komunikují a automaticky si vytváří i aktualizují informace ve svých směrových tabulkách. Ale v důsledku toho je mezi směrovači vyšší provoz (hlavně při startu, protože si směrovače potřebují předat všechny směrové informace o ostatních linkách), což spotřebovává šířku pásma. Rovněž jsou směrovače více vytěžovány, jak z hlediska paměti, tak i strojového času CPU a proto jsou na směrovače s dynamickým směrováním kladeny vyšší nároky, než u statického směrování.

V případě změny, či výpadku v síti, směrovač vypočítá a vybere jinou nejlepší cestu vedoucí k cíli. Důležitou roli tu hraje doba konvergence, což je v případě výpadku uzlu v síti čas potřebný pro konverzi směrového protokolu. Konvergence je opět dosažena v okamžiku, kdy je všem směrovačům v síti ohlášen daný výpadek a tato změna v topologii je zaznamenána do směrové tabulky u všech směrovačů.

V této úloze budeme využívat interních směrových protokolů, které se dělí do 2 základních skupin:

- **protokoly Distance Vector**,
- **protokoly Link State**.

1.1.4 Protokol RIP

Tento protokol patří mezi první protokoly dynamického směrování a spadá pod třídu Distance Vector protokolů, ze které vyplývají i jeho vlastnosti. K výpočtu metriky využívá pouze počtu přeskoků k cíli, kde dovoluje jen 15 přeskoků. Doba života paketu je totiž stanovena na 16. Síť vzdálená 16 a více přeskoků přes směrovače se označuje za nedosažitelnou (metrika je nastavena na nekonečno). RIP ke své funkčnosti používá 4 časovače (s defaultními hodnotami):

- **Update Timer** (30 sekund) – sděluje, jak často má být sousedním směrovačům posílána svoje celá směrová tabulka.
- **Invalid Timer** (180 sekund) – doba, do kdy je řádek ve směrové tabulce platný, pokud by od dané cesty nepřicházely žádné aktualizace. Po jejím vypršení je cestě přiřazena metrika 16, pokud tedy během odpočítávání směrovač neobdržel aktualizaci, v tom případě by byl časovač anulován. Po vypršení doby je směrovač ve stavu hold-down.
- **Hold-down Timer** (180 sekund) – Doba, po kterou je cesta přidržena. Směrovač v tomto časovém rozmezí rovněž nepřijímá žádné směrové aktualizace.
- **Flush Timer** (240 sekund) – Běží současně s časovačem Invalid Timer, tudíž po 60 sekundách, kdy byla cesta označena za neplatnou, je cesta vymazána z tabulky.

Tyto hodnoty časovačů musí být stejné na všech směrovačích v RIP síti, jinak je způsobena nestabilita sítě.

RIPv1 podporuje pouze třídní adresování (Classful) A, B a C (s prefixy /8, /16, /24). Při konfiguraci tedy není potřeba zadávat masku sítě, avšak tato metoda je velmi nevýhodná z hlediska špatného rozdělení počtu podsítí (třída A má k dispozici málo podsítí a třída C má naopak hodně podsítí). Tento protokol posílá aktualizační informace na všesměrovou adresu 255.255.255.255. Dnes se tato verze prakticky nepoužívá.

Velkým rozdílem oproti RIPv1 má **RIPv2** podporu třídního adresování (Classless), v sítích RIP tak můžeme používat proměnné délky masky podsítě (VLSM) a díky tomu sítě i efektivně sumarizovat, směrovač má pak uložené daleko méně adres než u RIPv1. Dále tento protokol podporuje autentizaci směrových informací s šifrováním MD5.

1.1.5 Protokol OSPF

Tento protokol spadá pod rodinu Link State protokolů a byl vyvinut především, aby překonal nedokonalosti Distance Vector, jako je tvorba nekonečných smyček, pomalá konvergence, či posílání celých tabulek, čímž se značně zatěžují právě velké sítě. Směrovače používající OSPF znají celou topologii sítě a udržují databázi stavů spojů. Rychle reagují na změny v síti. Každý směrovač hlídá dostupnost svých sousedů, ke kterým je připojen a v případě nějaké změny zasílá ihned jen danou informaci hierarchicky všem uzlům v síti. Aktualizace se zasílají pouze při změně, avšak minimálně jednou za 30 minut. Jako metriku využívá „cenu spoje“, která je ovlivňována šířkou přenosového pásma. K jejímu výpočtu je použit následující vzorec:

$$cena\ spoje = \frac{100\ \text{Mb}}{\text{šíř}ka\ pásma\ v\ \text{Mb}}.$$

Metrika cesty je součtem všech cen spojů na cestě k cíli a jako nejlepší cesta je označena ta s nejnižší hodnotou. Počet přeskoků k cíli metriku neovlivňuje, navíc u OSPF není počet přeskoků nijak limitován. Cenu spoje lze taky přepsat manuálně, kde pak můžeme upřednostnit pomalejší spoj nad rychlejším a tak zbytečně nezatěžovat hlavní spoj v případě

nízkých požadavků na přenos. K určení nejkratší cesty se používá Dijkstrův algoritmus.

Pro svou komunikaci používá OSPF 5 typů paketů:

- **Hello** – K sestavení vztahu sousednosti. Posílá se na základě 2 časovačů:
  - Hello interval – Slouží k udržení sousedství, posílá se každých 10 sekund.
  - Dead interval – Pokud po tuto dobu směrovač nepřijme žádný hello paket od sousedního směrovače, dojde k zániku sousedství. Posílá se každých 40 sekund (zpravidla 4 násobek hello intervalu).
- **Database Description** (DBD) – K přeposílání topologických informací.
- **Link State Request** (LSR) – K zažádání chybějící položky na základě obdrženého DBD paketu.
- **Link State Update** (LSU) – Slouží jako odpověď na LSR paket, obsahuje žádanou chybějící položku.
- **Link State Acknowledgement** (LSAck) – Potvrzuje přijetí LSU paketu.

Navázání vztahu sousednosti a výměnu směrových informací mezi směrovači popisuje následujících 7 stavů:

- **Down** – V tuto chvíli směrovače začínají komunikaci, doposud od sousedního směrovače nebyl obdržen Hello paket.
- **Init** – Směrovač obdržel Hello paket, ale obousměrná komunikace ještě není zřízena.
- **Two-Way** – Obousměrná komunikace je zřízena a v přijatém Hello paketu je v poli Neighbor: Router-ID vlastního směrovače. V této fázi se dále určuje tzv. pověřený směrovač (DR) a záložní pověřený směrovač, v případě, že jsou ve stejné síti připojeny více než 2 směrovače. Při změně v síti, pak všechny směrovače komunikují pouze přes pověřený směrovač a ne „každý s každým“.
- **Exstart** – Stanovení Master a Slave směrovače na základě velikosti sekvenčního čísla v prvotním odeslaném DBD paketu. Master poté zahajuje komunikaci.
- **Exchange** – Směrovače si vymění DBD pakety.
- **Loading** – Výměna LSR, LSU a LSAck paketů.
- **Full** – V tuto chvíli jsou směrovače plně synchronizované.

OSPF byl navrhnut především pro směrování ve velkých sítích, jelikož využívá tzv. hierarchického směrování, kdy je autonomní systém (AS) rozdělen na menší oblasti. Směrovače pak nemusejí znát všechny sítě v AS, ale pouze sítě ve své oblasti. Na hranici oblasti jsou adresy sítí sumarizovány.

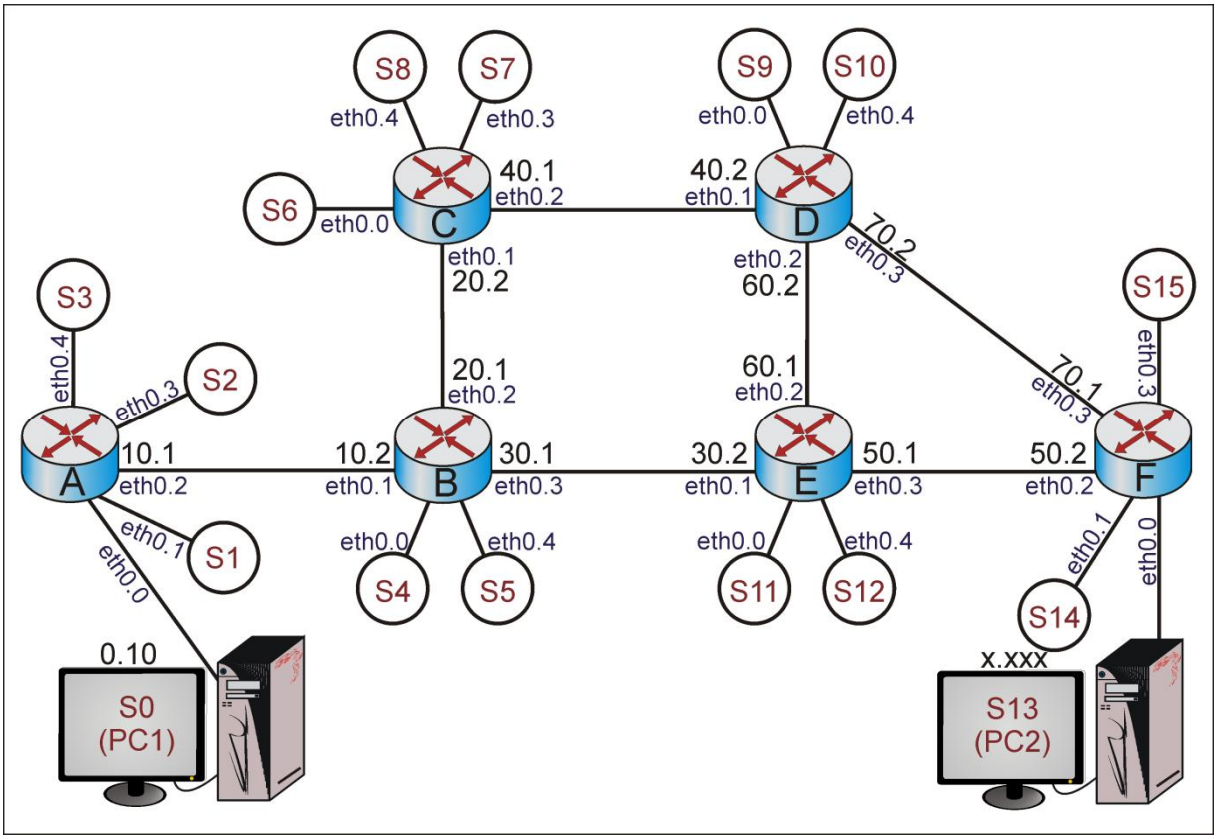
OSPF dále podporuje VLSM a jeho druhá verze **OSPFv2** umožňuje navíc i autentizace pomocí MD5.

1.2 Seznámení se s pracovištěm

V této laboratorní úloze budeme pracovat s 6 směrovači, ovšem jejich originální firmware nabízí velmi nízkou škálu možností konfigurace. Proto je zde použit firmware OpenWRT, který již nabízí možnosti konfigurace různých směrových protokolů. Veškeré směrování zajišťuje modul Zebra, který plní a udržuje v aktuálním stavu směrovou tabulku a dále se skrz tento modul konfigurují systémové záležitosti jako například název směrovače, rozhraní, přenosová šířka pásma a statické směrování. Dále jsou v OpenWRT přidané moduly pro konfiguraci protokolů RIP a OSPF. Pro možnosti nastavování těchto konfigurací je potřeba se nejdříve připojit na příslušný modul. Všechny tyto moduly naslouchají na adrese lokální smyčky (localhost, neboli 127.0.0.1) a na příslušném portu, kde modul Zebra naslouchá na portu 2601, RIP na portu 2602 a OSPF na portu 2604. Všechny moduly jsou

zabezpečeny stejným heslem: **zebra.** Po přihlášení k modulu je nám k dispozici velmi podobné terminálové rozhraní, jako je tomu u Cisco směrovačů.

Síť tohoto pracoviště má k dispozici adresový rozsah 10.10.0.0/16, v případě, kdy je nad portem napsané například 50.2 bude mít tento port IP adresu 10.10.50.2. Zapojení sítě je znázorněno na následujícím obr. 1 a adresace jednotlivých sítí v tab. 2. Všimněte si, že podsítě mezi směrovači mají vždy stejnou délku subnetu 30, zato sítě označené jako Sx pracují s proměnlivou délkou masky a jsou v rozsahu 10.10.0.0/21 (10.10.0.0 až 10.10.7.255).



Obr. 1: Zapojení pracoviště

Tab. 2: Konfigurační list

Síťový prvek	Rozhraní	IP rozhraní	IP sítě	Označení
RouterA	eth0.0	10.10.0.1	10.10.0.0/25	S0 (PC1)
	eth0.1	10.10.0.129	10.10.0.128/26	S1
	eth0.2	10.10.10.1	10.10.10.0/30	spoj A-B
	eth0.3	10.10.0.193	10.10.0.192/27	S2
	eth0.4	10.10.0.225	10.10.0.224/27	S3
RouterB	eth0.0	10.10.1.1	10.10.1.0/24	S4
	eth0.1	10.10.10.2	10.10.10.0/30	spoj A-B
	eth0.2	10.10.20.1	10.10.20.0/30	spoj B-C
	eth0.3	10.10.30.1	10.10.30.0/30	spoj B-E
	eth0.4	10.10.2.1	10.10.2.0/24	S5

Síťový prvek	Rozhraní	IP rozhraní	IP sítě	Označení
RouterC	eth0.0	10.10.3.1	10.10.3.0/25	S6
	eth0.1	10.10.20.2	10.10.20.0/30	spoj B-C
	eth0.2	10.10.40.1	10.10.40.0/30	spoj C-D
	eth0.3	10.10.3.129	10.10.3.128/26	S7
	eth0.4	10.10.3.193	10.10.3.192/26	S8
RouterD	eth0.0	10.10.4.1	10.10.4.0/24	S9
	eth0.1	10.10.40.2	10.10.40.0/30	spoj C-D
	eth0.2	10.10.60.2	10.10.60.0/30	spoj D-E
	eth0.3	10.10.70.2	10.10.70.0/30	spoj D-F
	eth0.4	10.10.5.1	10.10.5.0/24	S10
RouterE	eth0.0	10.10.6.1	10.10.6.0/25	S11
	eth0.1	10.10.30.2	10.10.30.0/30	spoj C-E
	eth0.2	10.10.60.1	10.10.60.0/30	spoj D-E
	eth0.3	10.10.50.1	10.10.50.0/30	spoj E-F
	eth0.4	10.10.6.129	10.10.6.128/25	S12
RouterF	eth0.0	10.10.7.1	10.10.7.0/25	S13 (PC2)
	eth0.1	10.10.7.129	10.10.7.128/26	S14
	eth0.2	10.10.50.2	10.10.50.0/30	spoj E-F
	eth0.3	10.10.70.1	10.10.70.0/30	spoj D-F
	eth0.4	10.10.7.193	10.10.7.192/26	S15

1.3 Postup řešení

1.3.1 Statické směrování

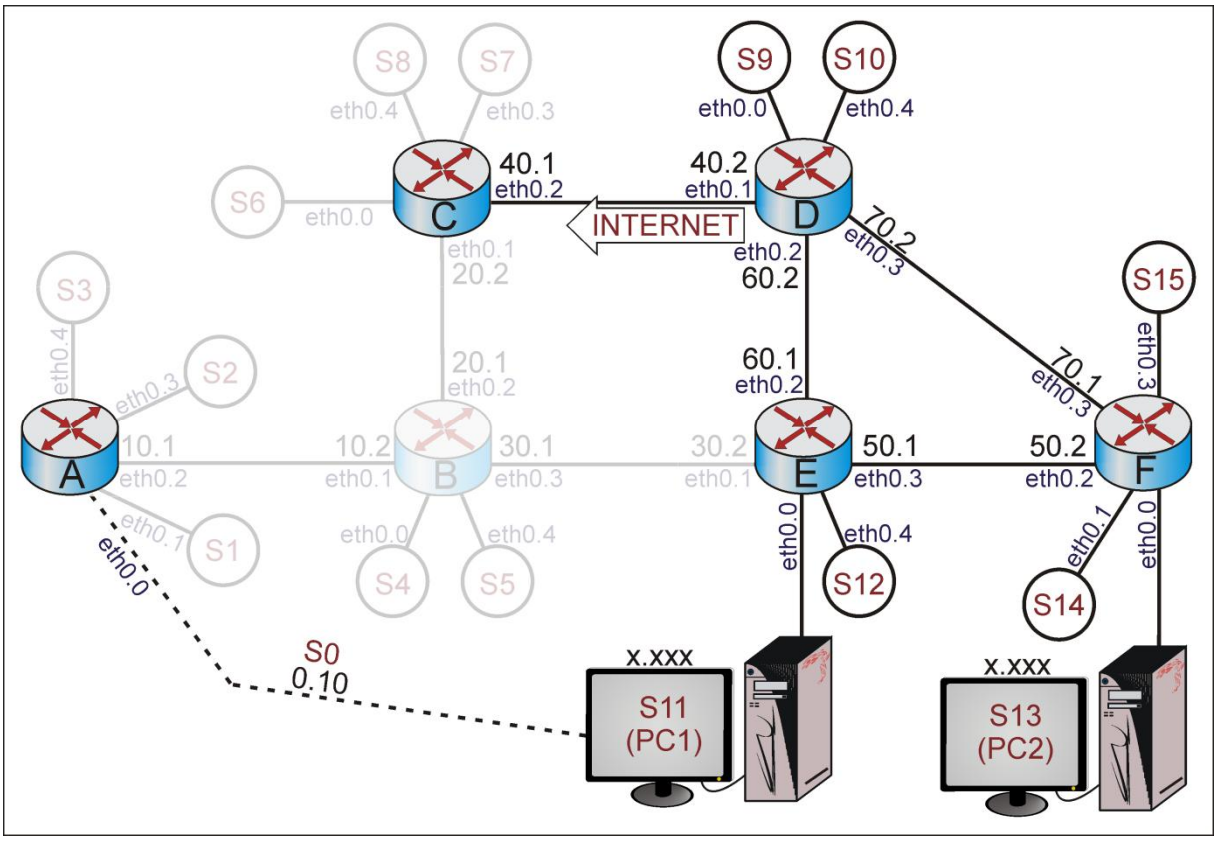
Po seznámení se s pracovištěm spusťte virtuální systémy v programu VirtualBox s názvem *BARS-Lx-PC1* a *BARS-Lx-PC2* (login: **student** a heslo: **student**). PC1 má na pozadí dostupné skripty důležité k přepisu konfigurací směrovačů. Nejprve je potřeba po předešlé skupině dát směrovače do původního stavu. POZOR: před spuštěním zkontrolujte, zda je PC1 (černý UTP kabel) zapojen v rozhraní **eth0.0** směrovače **RouterA** a měl přiřazenou statickou **IP adresu** z DHCP **10.10.0.10!!!** Ověření IP adresy proveďte příkazem **`ipconfig`** v příkazovém řádku **cmd** virtuálního systému, pokud přidělená adresa nesouhlasí a jste připojeni v eth0.0 směrovače RouterA, znamená to, že ještě nevypršela doba výpůjčky IP adresy z DHCP serveru, použijte tyto 2 příkazy (**`release`** k uvolnění IP adresy a **`renew`** k zažádání o novou IP adresu):

```
C:\Users\Student_BARS_1>ipconfig /release
C:\Users\Student_BARS_1>ipconfig /renew
```

Po chvíli budete mít přidělenou novou IP adresu počítače a výchozí bránu pro aktuálně připojenou síť. Během spuštění skriptu nepoužívejte myš ani klávesnici, vyčkejte na dokončení skriptu, který trvá přibližně 60 sekund!!! Po ověření IP adresy, spusťte skript **01_Static.vbs** umístěný na ploše. Skript postupně přepíše předešlé konfigurace, než budete dále pracovat, počkejte na dokončení.

Pro statické směrování budeme uvažovat zjednodušenou topologii sítě viz obr. 2, kde

budeme pracovat pouze se směrovači RouterD, RouterE a RouterF. Dále budeme předpokládat, že za směrovačem RouterD bude cesta vedoucí do Internetu (přes tento směrovač tedy chceme orientovat veškerý provoz z koncových sítí).



Obr. 2: Zjednodušená topologie pro statické směrování

Přepojte PC1 na RouterE, zažádejte si o novou IP adresu z DHCP a připojte se na směrovač pomocí *SSH* přes Putty (login: **root** a heslo: **linksys**). Následně se připojte do modulu Zebra příkazem:

`root@RouterE:~#` **`telnet localhost 2601`** (heslo: **zebra**).

Vstupte do privilegovaného režimu a poté do konfiguračního režimu pomocí příkazů:

`RouterE>`**`enable`**

`RouterE#` **`configure terminal`**

(doporučuji využívat příkazy zkráceně: **`en`** a **`conf t`**), následně můžeme přidávat statické záznamy do směrové tabulky, příkaz k tomu určený má tvar**:**

`ip route <cílová_síť>/<prefix> <next_hop>`,

next_hop je IP adresa dalšího skoku (taktéž se dá říct brána), tedy přes jakou IP adresu dalšího směrovače má paket putovat k *cílové_síti*. Cílem tohoto úkolu je sestavit spojení mezi všemi koncovými sítěmi a pro všechny neznámé vzdálené sítě trasovat cestu směrem na Internet, to můžeme provést několika způsoby. Jedna z možností je trasovat veškerý provoz ze směrovače RouterF na Internet přes směrovač RouterE. K tomuto nastavení využijeme statického záznamu v podobě implicitní cesty:

`RouterE(config)#` **`ip route 0.0.0.0/0 10.10.60.2`**

Směrovač RouterE, ale stále nezná sítě za směrovačem RouterF. Statické cesty k těmto sítím můžou vypadat následovně (složitý zápis, nepoužívejte):

`RouterE(config)#` **`ip route 10.10.7.0/25 10.10.50.2`**

```
RouterE(config)# ip route 10.10.7.128/26 10.10.50.2
RouterE(config)# ip route 10.10.7.192/26 10.10.50.2
```

Avšak tyto cesty lze efektivně sumarizovat. Zápis těchto cest může se stejným významem vypadat následovně:

```
RouterE(config)# ip route 10.10.7.0/24 10.10.50.2
```

Přepojte PC2 na RouterF, zažádejte si o novou IP adresu z DHCP a připojte se na směrovač pomocí *SSH* přes Putty (login: **root** a heslo: **linksys**). Následně se připojte do modulu Zebra příkazem:

```
root@RouterF:~# telnet localhost 2601
```
(heslo: **zebra**).

Vstupte do privilegovaného režimu a poté do konfiguračního režimu pomocí příkazů:

```
RouterF>enable
RouterF# configure terminal
```

(doporučuji využívat příkazy zkráceně: **en** a **conf t**), následně můžeme přidávat statické záznamy do směrové tabulky, příkaz k tomu určený má tvar**:**

ip route <cílová_síť>/<prefix> <next_hop>,

next_hop je IP adresa dalšího skoku (taktéž se dá říct brána), tedy přes jakou IP adresu dalšího směrovače má paket putovat k *cílové_síti*. Cílem tohoto úkolu je sestavit spojení mezi všemi koncovými sítěmi a pro všechny neznámé vzdálené sítě trasovat cestu směrem na Internet, to můžeme provést několika způsoby. Jedna z možností je trasovat veškerý provoz ze směrovače RouterF na Internet přes směrovač RouterE. K tomuto nastavení využijeme statického záznamu v podobě implicitní cesty:

```
RouterF(config)# ip route 0.0.0.0/0 10.10.50.1
```

Provoz procházející směrovačem RouterF je konečný, proto nám postačí tento jediný statický záznam. Příkazem *exit* se postupně odpojte z modulu Zebra do systému OpenWRT směrovače RouterF a připojte se opět na modul Zebra, avšak již na RouterD příkazem:

```
root@RouterF:~# telnet 10.10.70.2 2601
```
(heslo: **zebra**).

Nastavte implicitní trasování směrem na internet. Směrovač RouterD, ale stále nezná sítě za směrovačem RouterE. Statické cesty k těmto sítím můžou vypadat následovně (složitý zápis, nepoužívejte):

```
RouterD(config)# ip route 10.10.6.0/25 10.10.60.1
RouterD(config)# ip route 10.10.6.128/25 10.10.60.1
RouterD(config)# ip route 10.10.7.0/25 10.10.60.1
RouterD(config)# ip route 10.10.7.128/26 10.10.60.1
RouterD(config)# ip route 10.10.7.192/26 10.10.60.1
RouterD(config)# ip route 10.10.50.0/30 10.10.60.1
```

Avšak tyto cesty lze efektivně sumarizovat. Zápis těchto cest může se stejným významem vypadat následovně (složitý zápis, nepoužívejte):

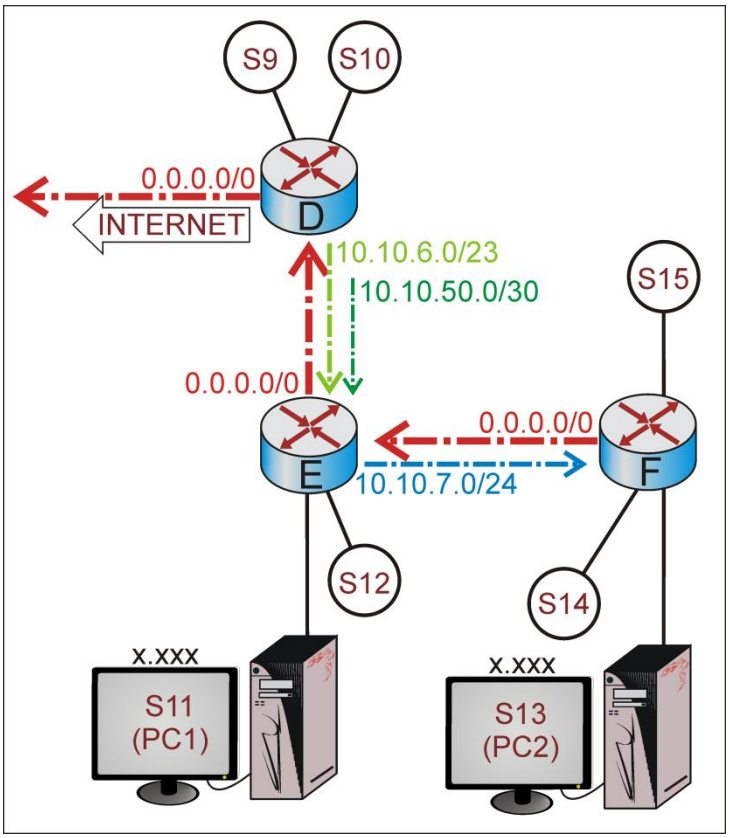
```
RouterD(config)# ip route 10.10.6.0/24 10.10.60.1
RouterD(config)# ip route 10.10.7.0/24 10.10.60.1
RouterD(config)# ip route 10.10.50.0/30 10.10.60.1
```

Cesty lze i dále sumarizovat:

```
RouterD(config)# ip route 10.10.6.0/23 10.10.60.1
RouterD(config)# ip route 10.10.50.0/30 10.10.60.1
```

Všimněte si, že na směrovači musíme nastavovat cesty vždy pro všechny sítě kromě přímo

přilehlých, jelikož přímo přilehlé sítě si směrovač zapisuje do směrové tabulky automaticky sám, ihned po nakonfigurování rozhraní.



Obr. 3: Nakonfigurované statické směrování

V tuto chvíli byste měli mít nakonfigurované cesty jako na obrázku 3. Spusťte nekonečný ping příkazem **`ping <IP_adresa> -t`** (ukončuje se pomocí klávesové kombinace *ctrl+c*), pokud je konektivita úspěšná, rozpojte spoj E-F, nyní je komunikace přerušena. Jelikož se jedná o statické směrování, další cestu k cíli musí opět nastavit administrátor. Upravte nastavení směrovačů tak, aby sítě S9 až S15 spolu opět navzájem komunikovaly. Příkazem **`no ip route <cesta>`** vymažte záznamy, které narušují správnost putování paketů (cestu na směrovači RouterD směřující pakety do sítě **`10.10.6.0/23`** není potřeba mazat!!). Pokud si nepamatujete vaši konfiguraci, můžete si ji zobrazit v privilegovaném režimu příkazem

```
RouterX# show running-config
```

Pamatujte, že pro komunikaci 2 bodů musí existovat cesta k cíli, ale i zpět. Využijte příkazu **`tracert <IP_adresa>`** v případě, kdy konektivita mezi počítači nebude správná. *Tracert* vám poskytne informaci, kam pakety proudí, popřípadě na kterém bodě se zastaví (paket nezná buď cestu dál, nebo cestu zpět z dalšího směrovače). Po správném sestavení spojení zapojte zpět spoj E-F.

1.3.2 Dynamické směrování – RIP

Před konfigurací protokolu RIP musíme vymazat statické záznamy, jelikož jejich administrativní vzdálenost je mnohem menší, tyto cesty by se proto vždy upřednostnily před dynamickým směrováním. Skript navíc nakonfiguruje protokol RIP na směrovače RouterC a RouterD. POZOR: před spuštěním skriptu zkontrolujte, zda je PC1 (černý UTP kabel) zapojen v rozhraní **eth0.0 směrovače RouterA** a měl přiřazenou statickou IP adresu z DHCP

10.10.0.10 (**`ipconfig release`** k uvolnění IP adresy a **`ipconfig renew`** k zažádání o novou IP adresu), poté spusťte skript **02_RIP.vbs**, během spuštění nepoužívejte myš ani klávesnici, vyčkejte na dokončení skriptu, který trvá přibližně 60 sekund!!!

Připojte se z PC1 na RouterA pomocí *SSH* přes Putty (login: **root** a heslo: **linksys**). Stejně jako jsme se předtím připojovali na modul Zebra, v této části úkolu se budeme vždy připojovat na modul RIP (port 2602):

`root@RouterA:~#` **`telnet localhost 2602`** (heslo: **zebra**).

V konfiguračním režimu zapněte směrování RIP příkazem:

`RouterA(config)#` **`router rip`**

specifikujte posílání a přijímání paketů pouze ve verzi 2 příkazem:

`RouterA(config-router)#` **`version 2`**

Nyní příkazem **`network <adresa_sítě>/<prefix>`** určíme, prostřednictvím jaké *sítě* budou přijímány a odesílány RIP aktualizace. Tyto aktualizace se posílají mezi směrovači, avšak v případě směrovače RouterA budeme posílat RIP aktualizace i na PC1, abychom měli možnost tento paket blíže prozkoumat. Zápis bude vypadat následovně (v průběhu práce doporučuji využívat na klávesnici tlačítko *up* v terminálu i příkazovém řádku, slouží k napsání předešle použitých příkazů):

`RouterA(config-router)#` **`network 10.10.0.0/25`**

`RouterA(config-router)#` **`network 10.10.10.0/30`**

Ostatním směrovačům musíme dát vědět i o dalších přilehlých sítích směrovače RouterA tím, že redistribuujeme tyto záznamy do směrového procesu RIP příkazem:

`RouterA(config)#` **`redistribute connected`**

Nyní je směrovač RouterA nakonfigurován, příkazem *exit* se odpojte z modulu RIP a připojte se opět na modul RIP, avšak již na RouterB příkazem:

`root@RouterA:~#` **`telnet 10.10.10.2 2602`** (heslo: **zebra**).

V konfiguračním režimu nastavte RouterB obdobně jako v předešlém případě.

Přepojte PC2 ke směrovači RouterF. Připojte se z PC2 na RouterF pomocí *SSH* přes Putty (login: **root** a heslo: **linksys**). Stejně jako jsme se předtím připojovali na modul Zebra, v této části úkolu se budeme vždy připojovat na modul RIP (port 2602):

`root@RouterF:~#` **`telnet localhost 2602`** (heslo: **zebra**).

V konfiguračním režimu zapněte směrování RIP příkazem:

`RouterF(config)#` **`router rip`**

specifikujte posílání a přijímání paketů pouze v RIP verzi 2 příkazem:

`RouterF(config-router)#` **`version 2`**

Nyní příkazem **`network <adresa_sítě>/<prefix>`** určíme, prostřednictvím jaké *sítě* budou přijímány a odesílány RIP aktualizace. Tyto aktualizace se posílají mezi směrovači. Zápis bude vypadat následovně (v průběhu práce doporučuji využívat na klávesnici tlačítko *up* v terminálu i příkazovém řádku, slouží k napsání předešle použitých příkazů):

`RouterF(config-router)#` **`network 10.10.50.0/30`**

`RouterF(config-router)#` **`network 10.10.70.0/30`**

Ostatním směrovačům musíme dát vědět i o dalších přilehlých sítích směrovače RouterA tím, že redistribuujeme tyto záznamy do směrového procesu RIP příkazem:

`RouterF(config-router)#` **`redistribute connected`**

Nyní je směrovač RouterF nakonfigurován, příkazem *exit* se odpojte z modulu RIP a připojte

se opět na modul RIP, avšak již na RouterE příkazem:

`root@RouterF:~#` **`telnet 10.10.50.1 2602`** (heslo: **zebra**).

V konfiguračním režimu nastavte RouterE obdobně jako v předešlém případě.

Ověřte komunikaci a určete trasu mezi sítěmi S0 a S13, využijte k tomu příkazu *tracert*. Na PC1 spusťte program Wireshark a odchyťte v něm komunikaci RIP protokolu. Než budete prozkoumávat obsah paketu RIP, spusťte *nekonečný ping* z PC2 na PC1 (**`ping 10.10.0.10 -t`**), odpojte spoj B-E a sledujte, za jakou dobu protokol RIP nalezne novou cestu k cíli. Jelikož nalezení nové cesty chvíli trvá, vrátíme se k odchycenému paketu RIP ve Wiresharku. Prozkoumejte blíže tento paket. Na jakém portu probíhá komunikace RIP protokolu? Jakou má tento paket velikost? Jak často se tento paket posílá? Zjištěné informace si poznamenejte, následně si projděte část s obsahem směrové tabulky a podívejte se na různé metriky jednotlivých sítí. Je již nová cesta nalezena? Jakou má nová cesta trasu? Zapojte linku B-E zpět.

1.3.3 Dynamické směrování – OSPF

Před konfigurací protokolu OSPF vymažte RIP záznamy ze směrové tabulky pomocí skriptu. Skript navíc nakonfiguruje protokol OSPF na směrovače RouterC a RouterD. POZOR: před spuštěním skriptu zkontrolujte, zda je PC1 (černý UTP kabel) zapojen v rozhraní **eth0.0 směrovače RouterA** a měl přiřazenou statickou IP adresu z DHCP **10.10.0.10** (**`ipconfig release`** k uvolnění IP adresy a **`ipconfig renew`** k zažádání o novou IP adresu), poté spusťte skript **03_OSPF.vbs**, během spuštění nepoužívejte myš ani klávesnici, vyčkejte na dokončení skriptu, který trvá přibližně 60sekund!!!

Připojte se z PC1 na RouterA pomocí *SSH* přes Putty (login: **root** a heslo: **linksys**) a dále se připojte na modul OSPF (port 2604):

`root@RouterA:~#` **`telnet localhost 2604`** (heslo: **zebra**).

V konfiguračním režimu zapněte směrování OSPF příkazem:

`RouterA(config)#` **`router ospf`**

Nastavte unikátní ID směrovače příkazem:

`RouterA(config-router)#` **`router-id 1.1.1.1`**

Definujte sítě, se kterými má směrovač RouterA udržovat stav sousedství příkazem **`network <adresa_sítě>/<prefix> area <číslo_oblasti>`**. Princip je stejný jako u protokolu RIP, avšak navíc zde specifikujeme do jaké OSPF oblasti bude směrovač patřit. V rámci této úlohy budou všechny směrovače v oblasti 0. Zápis konfigurace bude následující:

`RouterA(config-router)#` **`network 10.10.0.0/25 area 0`**

`RouterA(config-router)#` **`network 10.10.10.0/30 area 0`**

Redistribuujte do OSPF přímo přilehlé sítě:

`RouterA(config-router)#` **`redistribute connected`**

Nyní je směrovač RouterA nakonfigurován, příkazem *exit* se odpojte z modulu OSPF a připojte se opět na modul OSPF, avšak již na RouterB příkazem:

`root@RouterA:~#` **`telnet 10.10.10.2 2604`** (heslo: **zebra**).

V konfiguračním režimu nastavte RouterB obdobně jako v předešlém případě (unikátní ID již bude 2.2.2.2).

Přepojte PC2 ke směrovači RouterF. Připojte se z PC2 na RouterF pomocí *SSH* přes Putty (login: **root** a heslo: **linksys**) a dále se připojte na modul OSPF (port 2604):

`root@RouterF:~#` **`telnet localhost 2604`** (heslo: **zebra**).

V konfiguračním režimu zapněte směrování OSPF příkazem:

`RouterF(config)#` **`router ospf`**

Nastavte unikátní ID směrovače příkazem:

`RouterF(config-router)#` **`router-id 6.6.6.6`**

Definujte sítě, se kterými má směrovač RouterF udržovat stav sousedství příkazem **`network <adresa_sítě>/<prefix> area <číslo_oblasti>`**. Princip je stejný jako u protokolu RIP, avšak navíc zde specifikujeme do jaké OSPF *oblasti* bude směrovač patřit. V rámci této úlohy budou všechny směrovače v oblasti 0. Zápis konfigurace bude následující:

`RouterF(config-router)#` **`network 10.10.50.0/30 area 0`**

`RouterF(config-router)#` **`network 10.10.70.0/30 area 0`**

Redistribuujte do OSPF přímo přilehlé sítě:

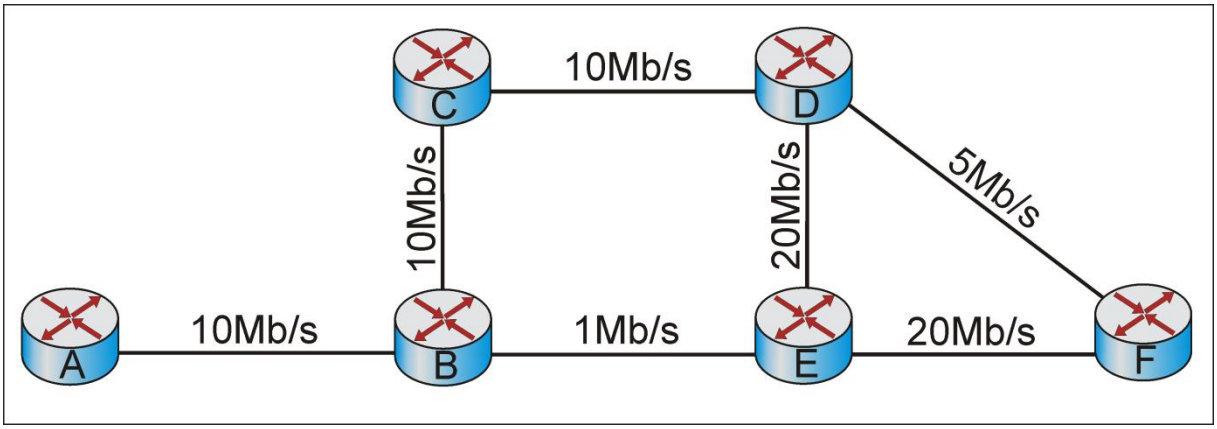
`RouterF(config-router)#` **`redistribute connected`**

Nyní je směrovač RouterF nakonfigurován, příkazem *exit* se odpojte z modulu OSPF a připojte se opět na modul OSPF, avšak již na RouterE příkazem:

`root@RouterF:~#` **`telnet 10.10.50.1 2604`** (heslo: **zebra**).

V konfiguračním režimu nastavte RouterE obdobně jako v předešlém případě (unikátní ID již bude 5.5.5.5).

Ověřte komunikaci a určete trasu mezi sítěmi S0 a S13, využijte k tomu příkazu *tracert*. Všimněte si, že cesta mezi počítači není stejná, jako tomu bylo u protokolu RIP, jelikož OSPF využívá k určení nejlepší trasy jinou metriku, která závisí na přenosové rychlosti spoje. Maximální přenosové rychlosti jsou již na směrovačích přednastaveny podle obr. 4. Z PC2 spusťte *nekonečný ping* na PC1 (**`ping -t 10.10.0.10`**), odpojení spoje B-E by v tuto chvíli nijak nezasáhlo do komunikace, proto odpojte spoj C-D a sledujte, za jakou dobu protokol OSPF nalezne novou cestu k cíli, porovnejte tento výsledek s protokolem RIP a opět zapojte spoj C-D.



Obr. 4: Maximální šířka pásma přenosu mezi směrovači

V privilegovaném režimu modulu OSPF na směrovačích RouterB, nebo RouterE si zobrazte OSPF databáze pomocí příkazu:

`RouterB#` **`show ip ospf route`**

Prohlédněte si metriky do vzdálených sítí. Blíže se podívejte v první tabulce na metriku sítě 10.10.30.0 (spoj B-E) a v druhé tabulce, přes která rozhraní jsou posílány pakety na vzdálené směrovače. Abychom spoj B-E zvýhodnili, musíme zvýšit jeho maximální šířku přenosového

pásma.

Z PC1 otevřete program Putty v novém okně a dvojklikem na profil RouterB se na něj přihlašte (login: **root** a heslo: **linksys**). Vstupte do modulu zebra příkazem:

```
root@RouterB:~# telnet localhost 2601
```
(heslo: **zebra**).

V konfiguračním režimu vstupte na rozhraní eth0.3 a nastavte přenosovou šířku pásma na 10 Mb/s:

```
RouterB(config)# interface eth0.3
RouterB(config-int)# bandwidth 10000
```

Z PC2 otevřete program Putty v novém okně a dvojklikem na profil RouterE se na něj přihlašte (login: **root** a heslo: **linksys**). Vstupte do modulu zebra příkazem:

```
root@RouterE:~# telnet localhost 2601
```
(heslo: **zebra**).

V konfiguračním režimu vstupte na rozhraní eth0.1 a nastavte přenosovou šířku pásma na 10 Mb/s:

```
RouterE(config)# interface eth0.1
RouterE(config-int)# bandwidth 10000
```

V privilegovaném režimu modulu OSPF na směrovačích RouterB, nebo RouterE si opět zobrazte OSPF databáze (**show ip ospf route)** a podívejte se, jak změna přenosové šířky pásma ovlivnila metriku cest a posílání paketů na vzdálené směrovače v druhé tabulce.

Na PC1 spusťte program Wireshark a odchyťte v něm komunikaci OSPF protokolu (zachytit lze pouze Hello paket, jelikož se počítač nechová jako směrovač a není schopen taktéž odpovědět paketem Hello, spojení mezi PC a směrovačem bude tedy pořád ve stavu *Down*). Prozkoumejte blíže tento paket. Jakou má velikost? Jak často se posílá? Porovnejte zjištěné informace s protokolem RIP. Který protokol více zahlcuje síťový provoz?

V rámci zbylého času můžete počítače různě přepojovat a určovat mezi nimi trasy putování paketů příkazem *tracert*.

1.4 Kontrolní otázky

1. V rámci prvního úkolu, jakou trasou by putoval paket s IP adresou cíle 10.55.1.1 ze sítě S12?
2. Zaměřte se na obrázek 2, na směrovači RouterD je nastavena cesta:
   ```
   RouterD(config)# ip route 10.10.6.0/23 10.10.60.1
   ```
   Proč nebylo nutné mazat tento statický záznam v prvním úkolu?
3. Jaké metriky u OSPF a RIP protokolu budou mít směrové záznamy na směrovači RouterE v případě, kdy budete chtít komunikovat mezi sítěmi S7 a S11.
4. Kolik paketů používá protokol OSPF ke své komunikaci?
5. Vyjmenujte výhody a nevýhody mezi statickým a dynamickým směrováním.

1.5 Seznam zkratek

AS	Autonomous System
DBD	Database Description
DHCP	Dynamic Host Configuration Protocol
DR	Designated Router
IP	Internet Protocol
LSAck	Link State Acknowledgement
LSR	Link State Request
LSU	Link State Update
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
SSH	Secure Shell
VLSM	Variable-Length Subnet Mask

1.6 Literatura

[1] BALCHUNAS, Aaron. Cisco CCNP Routing Study Guide. In: *Router Alley* [online]. Michigan, 2012 [cit. 2016-12-06]. Dostupné z: http://www.routeralley.com/completed/ccnp_routing_studyguide.pdf

[2] BOUŠKA, Petr. Cisco Routing 1 - obecné vlastnosti směrovacích protokolů. *Samuraj* [online]. 2009 [cit. 2016-11-02]. Dostupné z: http://www.samuraj-cz.com/clanek/cisco-routing-1-obecne-vlastnosti-smerovacich-protokolu

[3] GRYGÁREK, Petr. *Směrovací protokol OSPF*. Počítačové sítě [online]. Ostrava, 2016 [cit. 2017-03-25]. Dostupné z URL: http://www.cs.vsb.cz/grygarek/SPS/lect/OSPF/ospf.html

[4] MEDHI, Deepankar a Karthikeyan RAMASAMY. *Network routing: algorithms, protocols, and architectures*. San Francisco: Morgan Kaufmann Publishers, c2007. Morgan Kaufmann series in networking. ISBN 978-0-12-088588-6.z